REMARKS

Claims 1-47 are pending. Of those, claims 1, 24, 32 and 39 are independent.

RE-SUBMISSION, COPIES OF OCTOBER 3OTH IDS

The Examiner has indicated that the USPTO has no record of the Information Disclosure Statement (IDS) submitted by Applicant on October 30, 2001. As requested by the Examiner, Applicant is submitting copies of the October 30th IDS as well as proof of the submission thereof. To reduce the clerical burden, Applicant has not included copies of cited references that are U.S. Patents.

Accordingly, Applicant requests copies of the initialed forms PTO-1449 as confirmation that the references cited in the October 30th IDS have been made of record.

§ 102 Rejection

Beginning on page 2 of the Office Action, claim 24 is rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,654,752 to Ofek (the '752 patent). Applicant traverses.

From the Examiner's rebuttal remarks on p. 16 of the Office Action (again, mailed April 21, 2005), it seems as though the Examiner may not fully appreciate all that is recited by independent claim 24. To reiterate, independent claim 24 recites the following:

> (1) sending a first local mirror storage request from the local storage server across the first connection to a remote storage server, and
>
> (2) sending a first remote mirror storage request from the remote storage server across a second link to the local storage server.

Applicant is willing to assume for the sake of argument that sending of a first local mirror storage request is disclosed by the '752 patent.[1] But, as noted above, Applicant claims more.

A distinction of claim 24 over the '752 patent is sending a first remote storage request from the remote storage server to the local storage server. Again, nothing about the '752 patent suggests, much less discloses, that remote system 11 can send a remote mirror storage request to local system 10.

In view of the foregoing discussion, §102(e) rejection of claim 24 over the '752 patent remains improper and Applicant requests that it be withdrawn.

§ 103 Rejection, '752 Patent + '934 Patent

Beginning on page 4 of the Office Action, claims 1-22 and 25-43 are rejected under 35 U.S.C. 103(a) as being unpatentable over the '752 patent. Later, on p. 17 of the Office Action Office Action (again, mailed April 21, 2005), the Examiner modifies the rejection to supply U.S. Patent No. 6,417,934 to Sadr-Salek ("the '934 patent) as evidence that heartbeat signals in general were well known, which thus modifies the §103(a) rejection so as to be based upon a combination of the '752 patent as modified according to the '934 patent. Applicant continues to traverse.

Regarding independent claim 1, the Examiner acknowledges that the '752 does not teach the use of a heartbeat signal, but considers this to be taught by the '924 patent and further believes that the ordinarily skilled artisan would have modified the '754 patent according to the '934 patent. Applicant traverses.

[1] For brevity, Applicant has not repeated above the entire discussion of the '752 patent provided by the previous response. However, it is reiterated here that the entirety of the '572 patent treats remote system 11 as representing the mirror storage for local system 10. Remote system 11 never sends a mirror storage request to local system 10.

The heartbeat signal taught by the '934 patent arises in the context of communications between a data management server (DMS) 12 and a fax server (FS) 10. In most instances that a heartbeat signal is taught by the '934 patent, it is fax server 10 that sends a heartbeat signal to DMS 12 as a confirmation of having sent a fax. DMS 12 needs to receive such a confirmation before it can update its records; see col. 14, lines 18-21. As such, fax server 10 will keep sending the heartbeat signal until DMS 12 sends an acknowledgement signal (ACK) of having received the heartbeat signal.

Amended claim 1 recites, among other things, monitoring reception of at least one of the first heartbeat signal and the second heartbeat signal for interruption in the regular receipt thereof, respectively. Neither DMS 12 does this, nor does fax server 10. More particularly, DMS 12 does not monitor reception of heartbeat signals from fax server 10 for interruption in the regular receipt thereof. There is no such regular receipt taught by the '934 patent, hence there is nothing to be monitored for interruption. Even if regular receipt of a heartbeat signal was suggested by the '934 patent, DMS 12 is unconcerned with interruption. Rather, it's only responsibility is to send an ACK when a heartbeat signal is received from fax server 10. The same applies in the circumstance that DMS 12 sends a heartbeat signal to fax server 10.

Claims 2-22 depend at least indirectly from claim 1, respectively, and thus share its distinction over the applied combination of art.

Claims 25-31 depend at least indirectly from independent claim 24, respectively. A distinction of claim 24 over the '752 patent has been described above in the traversal of the §102(e) rejection. Claims 25-31 exhibit the same distinction by dependency.

Independent claims 32 and 39 recite features similar to those of claim 1, respectively, and thus similarly distinguish over the applied combination of art. Claims

33-38 and 40-43 depend at least indirectly from 32 and 39, respectively, and recite at least the same distinction as their base claims.

In view of the foregoing discussion, §102(e) rejection of claims 1-23 and 25-43 over the '752 taken alone is improper and Applicant requests that it be withdrawn.

§103 Rejection, '752 Patent & '587 Patent

Beginning on page 15 of the Office Action, claim 23 is rejected under U.S.C. 103(a) as being unpatentable over the '752 patent in view of U.S. Patent No. 6,633,587 to Bennett (the '587 patent).

Claim 23 depends indirectly from claim 1 and shares its distinctions noted above. The '587 patent has not been relied upon by the Examiner as a teaching corresponding to the claimed distinctions noted above by Applicant. Nor would it be reasonable to interpret the '587 patent as any such teaching.

In view of the foregoing discussion, the §103(a) rejection of claim 23 over the combination of the '752 and '587 patents is improper and Applicant requests that it be withdrawn.

<Remainder of page intentionally left blank.>

CONCLUSION

The issues in the case are considered to be resolved. Accordingly, Applicants request a Notice of Allowability.

Person to Contact

In the event that any matters remain at issue in the application, the Examiners are invited to contact the undersigned at (703) 668-8000 for the purpose of a telephonic interview.

If necessary, the Commissioner is hereby authorized in this, concurrent, and future replies, to charge payment or credit any overpayment to Deposit Account No. 08-2025 for any additional fees under 37 C.F.R. §§ 1.16 or 1.17; particularly, extension of time fees.

Respectfully submitted,

Nicos A. Vekiarides

By: Thomas S. Auchterlonie
Thomas S. Auchterlonie
Reg. No. 37,275

HARNESS, DICKEY & PIERCE, P.L.C.
P.O. Box 8910
Reston, VA 20195
(703) 668-8000

TSA/tsa

Enclosures:
Copy of October 30, 2001 IDS, sans copies of U.S. Patent references
Proof that IDS submitted October 30th (copy of stamped postcard)

AL 1

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G06F 11/20, 11/14	A1	(11) International Publication Number: WO 92/18931 (43) International Publication Date: 29 October 1992 (29.10.92)

(21) International Application Number: PCT/US92/03001

(22) International Filing Date: 14 April 1992 (14.04.92)

(30) Priority data:
690,066 23 April 1991 (23.04.91) US

(71) Applicant: EASTMAN KODAK COMPANY [US/US]; 343 State Street, Rochester, NY 14650 (US).

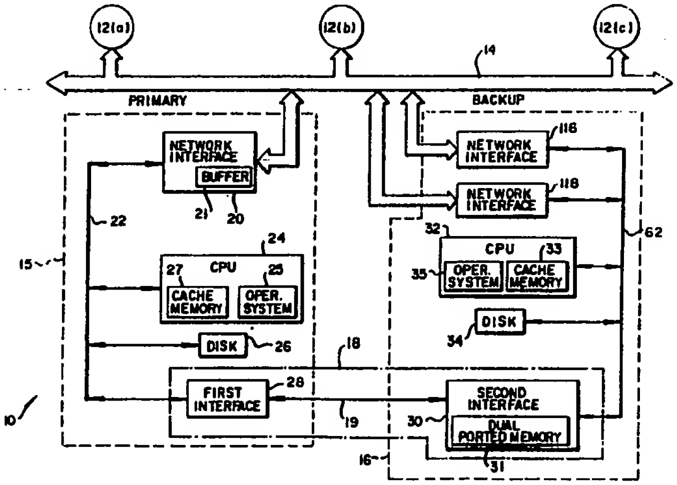
(72) Inventors: VINTHER, Gordon ; 22 Jersey Street, Pepperell, MA 01463 (US). McGRATH, James, W. ; 108 Kinnaird Street, Cambridge, MA 02139 (US).

(74) Agent: DUDLEY, Mark, Z.; 343 State Street, Rochester, NY 14650-2201 (US).

(81) Designated States: AT (European patent), BE (European patent), CH (European patent), DE (European patent), DK (European patent), ES (European patent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, LU (European patent), MC (European patent), NL (European patent), SE (European patent).

Published
With international search report.

(54) Title: FAULT TOLERANT NETWORK FILE SYSTEM



(57) Abstract

A fault tolerant network fileserver system includes a plurality of nodes connected to a network communication link. A primary fileserver node stores files from a plurality of the nodes and a backup fileserver node stores copies of files from the primary fileserver. In an improved fileserver system, the primary and backup fileservers are connected to a dual ported memory for communicating information between the fileservers. The primary fileserver writes data files to the dual ported memory and interrupts a processor within the backup fileserver to notify it that the dual ported memory contains data. In response to the interrupt, the processor within the backup fileserver reads the data from the dual ported memory and writes it to a storage device within the backup fileserver. In a similar manner, the dual ported memory is used for passing control messages between the primary and backup fileservers. The dual ported memory includes semaphore locations for arbitrating between competing requests by the backup and primary fileservers for access to the same location in the dual ported memory.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FI	Finland	ML	Mali
AU	Australia	FR	France	MN	Mongolia
BB	Barbados	GA	Gabon	MR	Mauritania
BE	Belgium	GB	United Kingdom	MW	Malawi
BF	Burkina Faso	GN	Guinea	NL	Netherlands
BG	Bulgaria	GR	Greece	NO	Norway
BJ	Benin	HU	Hungary	PL	Poland
BR	Brazil	IE	Ireland	RO	Romania
CA	Canada	IT	Italy	RU	Russian Federation
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland			SN	Senegal
CI	Côte d'Ivoire	KR	Republic of Korea	SU	Soviet Union
CM	Cameroon	LI	Liechtenstein	TD	Chad
CS	Czechoslovakia	LK	Sri Lanka	TG	Togo
DE	Germany	LU	Luxembourg	US	United States of America
DK	Denmark	MC	Monaco		
ES	Spain	MG	Madagascar		

FAULT TOLERANT NETWORK FILE SYSTEM

BACKGROUND OF THE INVENTION

This invention relates to a fault tolerant network file system having a primary fileserver and a backup fileserver which mirrors the primary. When the primary fails, the backup assumes the role of the primary on the network in a manner transparent to users whose files are stored on the primary.

A network generally includes a group of nodes which communicate with each other over a high speed communications link. Examples of nodes include single user personal computers and workstations, multiple user computers, and peripheral devices such as image scanners, printers and display devices.

Many networks include a fileserver node which operates as a central storage facility for data and software used by many nodes on the network. The fileserver includes a disk storage device, for storing the data and software, and a central processing unit (CPU) for controlling the fileserver. Files arriving at the fileserver over the communication link are typically first stored in a cache memory within the central processing unit and later copied to the disk storage device for permanent storage.

If the fileserver fails before the files in cache memory are copied to the disk, the files may be irretrievably lost. Accordingly, a client node sending certain critical files to the fileserver may instruct the fileserver to immediately write specified files to disk, thereby reducing the likelihood that the specified files will be lost in the event of such a failure.

Since many nodes rely on the fileserver, many users will be affected if the fileserver fails. For example, even if all files are safely stored on the primary fileserver's disk, a temporary failure of the primary fileserver will inconvenience many users since their data files are unavailable until the fileserver is restored to service. Networks which require a high degree of availability typically emplc̈y techniques to hedge against such failure. For example, a backup fileserver may be used to maintain a copy of the primary's files. Upon failure of the primary, an exact copy of the primary's files can be automatically accessed through the backup fileserver.

One object of the invention is to provide a backup fileserver which promptly mirrors each change of the primary's files. Thus, when the primary fails, the files of the backup fileserver match those of even the most recently added or modified files of the primary. A further object of the invention is to achieve this mirroring at high speed and with little computational cost. Yet another object is to provide the elements for switching between the primary fileserver to the backup fileserver in a manner transparent to the users.

SUMMARY OF THE INVENTION

The invention relates to an improved fault tolerant network fileserver system. The network fileserver system includes a network communication link connected to a plurality of nodes. A primary fileserver and a backup fileserver are also connected to the network communication link for storing files from the nodes.

In the improved fileserver system, the primary fileserver includes a primary computer processor, a primary storage disk, a first network interface connected to the network communication link, and a first independent interface connected to the backup fileserver. The first independent interface is responsive to commands from the primary processor for communicating information to the backup fileserver.

The backup fileserver includes a backup computer processor, a backup storage disk, a backup network interface connected to the network communication link, and a second independent interface connected to the first independent interface. The second independent interface includes a dual ported memory. It receives information from the primary fileserver and stores the information in the dual ported memory. In response to commands from the backup processor, the second independent interface provides information from the dual ported memory.

In preferred embodiments, the second independent interface includes a means for interrupting the backup computer processor to notify it that the dual ported memory contains information received from the primary fileserver. A portion of the dual ported memory is arranged as a group of data memory blocks, each for storing data to be copied to the backup storage disk. Other portions store an entry pointer identifying a next available data block, and a removal pointer identifying a next data block to be emptied.

Other portions of the dual ported memory may be arranged as a first control message block, for storing control messages from the primary fileserver to the backup fileserver, and a second control message block, for storing control messages from the backup fileserver to the primary fileserver.

The second independent interface includes at least one common register accessible to both the primary and backup processors for indicating the entry and removal of data blocks from the data block portion. For example, a preferred embodiment includes a count register. The second interface includes a means responsive to signals from the first independent interface for modifying the contents of the count register to indicate the entry of data into a memory block. The second independent interface is further responsive to signals from the backup computer processor for modifying the count register to indicate the removal of data from a memory block.

The dual ported memory includes a semaphore register for arbitrating access to the count register. The semaphore register has a first and a second state of operation. When read by a processor while in the first state, the semaphore register provides a register available code and automatically enters the second state. When read by a processor while in the second state, the semaphore register provides a register unavailable code and remains in the second state. When written to by a processor while in the second state, the semaphore register returns to the first state.

In the preferred embodiment, the primary fileserver further includes an improved Unix operating system for controlling the copying of files received by the first network interface to the primary storage disk and to the dual ported memory. Conventional Unix operating systems receive both secure disk write instructions and unsecure disk write instructions directing the operating system to write specified files to the primary storage disk. In response to an unsecure write instruction, an unsecure write procedure writes to the primary storage disk using an efficiency algorithm which temporarily defers copying the files to disk. In response to a secure write instruction, a secure procedure writes files to disk promptly without the benefit of the efficiency algorithms. The improved operating system responds to both secure and unsecure instructions by promptly writing the specified files to the dual ported memory, and subsequently writing the specified files to the primary disk storage device using the unsecure write procedure.

Other objects, features and advantages of the invention are apparent from the following description of a preferred embodiment taken together with the drawings.

Brief Description of the Drawings

Figure 1 is a block diagram of a computer network having a primary fileserver and a backup fileserver.

Figure 2 is a block diagram of a pair of interface boards for connecting the primary fileserver to the backup fileserver.

Figure 3 is a diagram of several registers on the interface boards shown in Figure 2.

Figure 4 is a diagram showing the organization of a dual ported memory according to a preferred embodiment of the invention.

Figures 5(a) and 5(b) are a flow chart of a method for copying new files of the primary fileserver to the dual ported memory.

Figures 6(a) and 6(b) are a flow chart of a method for copying data from the dual ported memory to a backup fileserver.

Figures 7(a) through 7(d) are a flow chart of a procedure by which a backup fileserver temporarily replaces the primary fileserver.

Figures 8(a) and 8(b) are a flow chart of a procedure by which a failed primary fileserver resumes its role on the network after recovering from a failure.

Description of the Preferred Embodiment

Referring to Fig. 1, a network 10 includes a plurality of nodes 12, each for performing specific tasks in the design and production of publications such as newspapers and magazines. For example, node 12(a) could be a computer workstation which allows a journalist to draft an article to be included in the publication; node 12(b) could be a scanner for digitizing an image, such as a photograph, to be printed with the article prepared at node 12(a); and node 12(c) could be a computer workstation which allows a user to lay out the entire publication by selecting and arranging articles, images and advertisements prepared by other nodes on the network.

The network may include other nodes for producing hard copies of the publication. For example, printer nodes prepare hard copies of images called "proofs". Other nodes prepare printing plates used in high volume printing of the publication.

Each node 12 communicates with other nodes via a high speed communication link 14. For example, in networks conforming to the Ethernet protocol, link 14 is a high speed serial communication channel over which a node can broadcast messages to one or more other nodes. Many nodes include disks for storage of information used locally on the node. For example, workstations 12(a) and 12(c) typically include disks for storing software used in performing their respective tasks, e.g., text editing and document layout.

However, many nodes are diskless and therefore require access to a central fileserver 15 for storing information and accessing software. Further, even nodes having disks often store data on the central fileserver. Requiring nodes to store data files in a central fileserver provides several advantages including the centralized control of a common file system which is accessible to many nodes on the network.

Failure of the fileserver node is catastrophic since the fileserver contains information needed by a large number of nodes. Accordingly, a backup fileserver 16 is connected to the primary fileserver through a parallel port 18 to maintain a copy of all of the primary's files. Upon failure of the primary, the backup automatically assumes the role of the primary in a manner transparent to the other nodes on the network. More specifically, other nodes may continue to access the central filesystem without any special instruction from the users.

The primary fileserver includes a first network interface 20 for receiving messages (e.g., files) from communications link 14 and storing them in a buffer memory 21. A central processing unit CPU 24 responds to instructions from operating system software 25 to transfer the buffer contents over a parallel bus 22, through a first interface 28, across a cable 19 to a second interface 30 within backup fileserver 16. Second interface 30 includes the dual ported memory 31 which temporarily stores the buffer data.

After storing data in memoryy31, CPU 24 instructs interface 30 to interrupt a CPU 32 within backup fileserver 16, thereby notifying CPU 32 that memory 31 includes data. In response to the interrupt, CPU 32 moves the buffer data contents of dual ported memory 31 to a cache memory 33. An operating system 35 later instructs CPU 32 to copy the data from cache 33 to the backup disk 34, thereby providing disk 34 with a copy of each file arriving from the network.

After CPU 24 has copied the buffer data to the dual ported memory, the operating system 25 instructs CPU 24 to move the data to a data cache memory 27. Operating system 25 next instructs CPU 24 to copy the contents of cache memory 27 to a primary disk 26. As explained below, the operating system provides three types of write procedures for copying the files from cache 27 to disk 26.

In the preferred embodiment, operating system 25 is a Unix operating system, modified to include code for interacting with backup fileserver 16 to copy files from buffer memory 21 to dual ported memory 31. Conventional Unix operating systems include filesystem code for maintaining a system of data files on a disk.

The filesystem code includes several types of write procedures for copying files from cache 27 to disk. For files created locally on the primary fileserver (i.e., files which do not arrive over communication link 14), the filesystem code typically employs a delayed write procedure. The delayed write procedure implements efficiency algorithms which manage the writing of data to disk to avoid unnecessary delays. For example, the delayed write procedure searches the cache for files which are assigned to the same disk "cylinder". It then instructs the CPU to initialize a direct memory controller (DMA) to successively transfer the selected files to disk. After this initialization is complete, the CPU returns to other tasks while the DMA controller attends to copying the selected data files to disk. Since the files are stored in the same disk cylinder, a time consuming "cylinder seek" operation is avoided by writing these files successively.

For files arriving over link 14, a "synchronous" write procedure is typically used. The synchronous write procedure promptly writes the file to disk without employing the efficiency algorithms described above. By promptly removing the file from the volatile cache, the synchronous write procedure reduces the likelihood that the file contents will be lost in the event of a failure of the fileserver. However, this procedure degrades disk performance by preventing the efficiency algorithms from reducing the number of time consuming cylinder seek operations.

Further, the client server may also operate in a synchronized fashion, waiting for the fileserver to confirm that the file has been written to disk before proceeding to execute the application program which prompted the write to the filesystem. Such "remote synchronous" writes provides a high degree of security against a failure of the fileserver since the application program on the client node cannot proceed until the data is safely stored on disk. However, this procedure degrades performance of the client node since the application program is stalled for a long period of time waiting for the relatively slow disk write procedure to complete and for the acknowledge message to arrive over the relatively slow network.

Of the above types of disk writes, the relatively unsecure delayed disk write procedure is the least taxing on the system performance. However, this procedure risks the loss of crucial data. In the event of a failure of the fileserver, contents of the volatile cache memory are destroyed. Accordingly, delayed writes are typically used for relatively unimportant data while synchronous and asynchronous writes are used for relatively critical data.

Backup fileserver 16 provides a high degree of security against a failure without the need for the costly synchronous disk writes. More specifically, each file is initially copied to the dual ported memory 31 which is powered and controlled by the backup fileserver. Since the backup fileserver is thus independent of the primary, there is no need to immediately copy the volatile cache memory 27 to the primary's nonvolatile disk 26. Accordingly, operating system 25 is a modified Unix operating system wherein all synchronous disk writes are converted to delayed writes. By eliminating synchronous writes, the performance of the fileserver is dramatically improved since the efficiency algorithms coordinate all disk writes to optimize disk performance. Further, client nodes running application programs which call for many remote synchronous writes (such as document processing applications of the preferred embodiment) receive an acknowledgment from the fileserver as soon as the file has been written to the relatively fast dual ported memory. Thus, the performance of the client node is dramatically improved since it need no longer wait for the slow disk write procedure.

OPERATION OF DUAL PORTED MEMORY

Referring to Fig. 2, dual ported memory 31 includes a nineteen bit address bus ADDR which is accessible to both CPU 24 and CPU 32. To access a given location in memory 31, CPU 24 first loads a pair of address registers 40, 42 with the address of the location to be accessed. As shown in Fig. 3, register 42 contains the low order sixteen bits of the address (i.e., bits A0 - A15) and register 40 contains the upper three bits of the address (bits A16 - A18). When CPU 24 performs a read or write cycle directed to memory 31, the contents of registers 40, 42 are automatically applied to the address bus ADDR, thereby pointing to a specific location within memory 31.

To load address register 40, CPU 24 performs a write cycle to a predetermined address assigned to the register. An address decoder 44, within interface 28, decodes the address from parallel bus 22. Upon recognizing the predetermined register address, decoder 44 asserts a pair of encoded control signals "REG" which indicate that CPUý24 is requesting access to register 40. A second address decoder 46, within interface 30, decodes the control signals and asserts a register enable signal R1 which selects register 40. The read/write control signal from bus 22, which indicates that a write is being performed, is forwarded by address decoder 44 to interface 30. Decoder 46 receives the forwarded R/W signal and provides a corresponding buffered signal "R/W1" to register 40. Since the buffered read/write signal R/W1 indicates that a write is being performed, the activation of register enable signal R1 causes register 40 to load the data from data bus DB into the register cells.

The data on bus DB is provided by CPU 24 through transceivers 50, 52. More specifically, address decoder 44 monitors the read/write control signal from bus 22. Upon recognizing a write cycle to a location on interface 30, decoder 44 enables transceiver 50 to drive data from bus 22 across cable 19. Similarly, when decoder 46 recognizes an access to interface 30, it enables transceiver 52 to forward the data from cable 19 to data bus DB.

In the same manner, CPU 24 loads register 42 by performing a write to an address assigned to register 42. In response to this write cycle, decoder 46 asserts a second register enable signal R2 causing the data from bus DB to be loaded into register 42.

Once registers 40, 42 are initialized with the appropriate address, CPU 24 writes data to the addressed location in memory 31 by performing a write cycle to a predetermined address assigned to memory 31. In response, decoder 46 asserts memory register signal R3 which instructs registers 40, 42 to assert the stored address on the memory address bus ADDR. The memory register signal R3 is further provided to a multiplexer 54 which in response, applies a memory access signal "MEM-Select" to memory 31. The buffered read/write control signal R/W1 is also applied to multiplexer 54 which in response asserts a memory read/write control signal "MEM-R/W". Since MEM-R/W indicates that the present cycle is a write cycle, memory select signal "MEM-Select" instructs memory 31 to load the data D0-D16 (see Fig. 3) from data bus DB to the location provided on the address bus ADDR. Thus, the predetermined address assigned to memory 31 behaves as a sixteen bit register 43 whose contents are determined by the contents of the memory location pointed to by address registers 40, 42.

Referring to Fig. 3, register 40 is a sixteen bit register. However, only five bits in the register are used. As explained above, the low order three bits store the high order address bits (A16 - A18). Bit nine is a write increment bit "WI1". If this bit is set, the address in registers 40, 42 is incremented each time CPU 24 writes to a location in memory 31. Thus, CPU 24 can write to a block of locations in memory 31 by loading registers 40, 42 with the base address of the block and setting the write increment bit. CPU 24 then simply repeatedly writes data to the address dedicated to memory 31. Registers 40, 42 increment the memory address with each write, thereby loading successive locations in memory 31. CPU 24 can similarly read a block of memory by setting read increment bit RI1 in register 40 and successively reading from memory 31.

CPU 32 reads and writes data from memory 31 in an analogous manner. For example, to read a block of data from memory 31, CPU 32 loads a pair of address registers 56, 58 with the address of the first location to be read, and sets read increment bit RI2 in register 56. It next reads data from the predetermined address dedicated to memory 31. An address decoder 60 decodes the address from bus 62 and upon recognizing the address of memory 31, asserts a memory access signal R7 instructing address registers 56, 58 to assert the stored address on address bus ADDR. Access signal R7 is also applied to multiplexer 54 which responds by asserting memory access signal "MEM-Select". In response to access signal MEM-Select, memory 31 provides data D0-D15 from the memory location identified by the address on bus ADDR to data bus DB. Thus, from the perspective of CPU 32, the predetermined address assigned to memory 31 behaves as a sixteen bit register 43 whose contents are determined by the contents of the memory location pointed to by address registers .

Upon recognizing a read cycle from CPU 32 directed to a location on interface 30, decoder 60 asserts an "enable" signal causing a data transceiver

64 to assert the contents of data bus DB onto bus 62, thereby providing CPU 32 with the desired data. When the read increment bit is set, registers 56, 58 automatically increment the stored address. Thus, CPU 32 reads the next location in memory 31 by again reading from the address assigned to memory 31.

Referring to Figs. 4, 5(a), 5(b), and 6 the following describes in more detail the software procedure for moving data from the primary fileserver 15 to the backup fileserver 16.

ORGANIZATION OF DUAL PORTED MEMORY

Fig. 4 illustrates the memory map for dual ported memory 31 of the preferred embodiment. Memory 31 includes a semaphore block 70 at the beginning of memory (i.e. the first 512 locations) which contain semaphores used in controlling access to certain registers to be described below. The semaphore block is followed by a control register block 72 containing various registers used by CPU 24 and CPU 32 in passing data and control messages as described below.

Control register block 72 is followed by a pair of control message blocks 74, 76. The first control message block 74 is used by CPU 24 to send control messages to CPU 32. Each message is defined by three words 75. To send a message, CPU 24 writes the corresponding three word code to a first container 78 within the message block 74. Subsequent messages are written to adjacent containers.

CPU 32 removes the messages on a first-in-first-out basis. By the time CPU 24 loads the last container 80 in the block, CPU 32 has already emptied the first. Thus, after loading the last container, CPU 24 returns to the first container. The control message block 74 thus operates as a circular buffer.

Control message block 76 is used in the same manner to transfer control messages from CPU 32 to CPU

24. CPU 32 loads messages into the circular buffer and CPU 24 removes them on a first-in-first-out basis.

Control message blocks 74, 76 are followed by a group of data memory blocks 82. The data memory blocks are used to transmit data between CPU 24 and CPU 32 in the same manner that control message blocks 74, 76 are used to transmit control messages. More specifically, CPU 24 loads a first block of data into a first memory block 84. As each new block of data arrives, CPU 24 loads the data into the next memory block.

As CPU 24 loads data blocks into memory, CPU 32 removes them on a first-in-first-out-basis. By the time CPU 24 loads the last memory block 86, CPU 32 has already emptied the first memory block 84. Thus, once CPU 24 loads the last memory block 86, it returns to the first memory block 84. Memory blocks 72 therefore operate as a circular buffer.

OPERATION OF CIRCULAR BUFFERS WITHIN DUAL PORTED MEMORY

Referring to Figs. 5(a) and 5(b), the operation of the circular buffers are now described in more detail, using as an example the transfer of data blocks.

Control register block 72 includes a next entry pointer 88 which points to the next available data block in memory 31. It also includes a removal pointer 90, which points to the next data block to be emptied, and a count register 92 which specifies the number of data blocks currently stored in memory 31.

When CPU 24 desires to load a block of data to memory 31, it first reads count register 92 to determine whether the circular buffer is full. (Step 210). CPU 32 typically removes blocks quickly enough that the circular data buffer should never become full. However, if the buffer becomes full, CPU 24 repeatedly reads the count register 92 until CPU 32 frees a block and decrements the count. (Step 212).

If the buffer is not full, CPU 24 reads the entry pointer 88 to determine the location of the next available data block. (Step 214). It then initializes registers 40, 42 with the address of the first location in the selected block and sets the write increment bit WI in register 40. (Step 216). CPUÿ24 then loads data into the selected memory block by performing successive writes to memory 31. (Step 218). When the block of data is loaded, CPUÿ24 updates the entry pointer 88 to point to the next available block. (Step 220).

Finally, CPU 24 must increment the count to indicate that a new block has been added. However, CPU 32 may also attempt to modify the count at the same time, i.e., to decrement the count to reflect the removal of a data block. For example, assume both CPU 24 and CPU 32 read the current value of the count which is five. CPU 24, having just loaded a new data block, seeks to increment the count to six. CPU 32, having just removed a block, seeks to decrement the count to four. If no mechanism is provided to synchronize CPUÿ24 with CPUÿ32, CPUÿ24 may write a six to the count register 92 and CPUÿ32 will overwrite this value with a four. Yet the count should remain at five since five blocks remain in memory 31.

To avoid this type of error, a count semaphore word 96 is stored in semaphore blockÿ70. (Fig. 4). Before CPUÿ24 reads the count register 92, it first reads count semaphore 96. (Step 222). If the semaphore is zero, (Step 224) CPUÿ24 assumes it has control over the countÿregister.

When CPU 24 reads a zero from semaphore 96, memoryÿ31 automatically sets the semaphore to one, thereby indicating to CPUÿ32 that countÿregister 92 is under the exclusive control of CPUÿ24. After CPUÿ24 reads the count and increments it (Steps 226, 228), CPU 24 writes a zero to semaphore 96, thereby freeing the countÿregister 92 for use by CPUÿ24. (Step 230).

If the count read in step 226 was zero, thereby indicating that the circular buffer was empty before CPU 24 added the last block, CPU 24 interrupts CPU 32 to notify it that the buffer now contains data. (Steps 232, 234). (The mechanism by which CPU 24 interrupts CPU 32 will be described in detail below.) If the count is greater than zero, an interrupt should already be pending due to the previously loaded data block which has not yet been removed. Accordingly, CPU 24 returns to other operations. (Step 236).

Referring to Figs. 6(a) and 6(b), CPU responds to an interrupt by first clearing the interrupt as described below. (Step 310). It proceeds to remove a block of data from memory 31 by first reading the removal pointer 90 to determine the location of the next block in the buffer. (Step 311). CPU 32 next initializes registers 56, 58 with the address of the first location in the block to be emptied and sets the read increment bit RI2 in register 56. (Step 312). It then empties the block by successively reading from memory 31 and writing the data to cache 33. (Step 314). When the entire block is emptied, CPU 32 updates the removal pointer 90 to point to the next block in the buffer. (Step 316).

Finally, CPU 32 updates the count register 92 to reflect the removal of a block from the buffer. Toward this end, it first reads the count semaphore 96. (Step 318) If the semaphore is set to one, indicating that CPU 24 has control of the count, CPU 32 repeatedly reads the semaphore until it returns to a zero. (Step 320) Once the semaphore clears, CPU 32 reads the count from register 92 and decrements register 92 to reflect the removal of a block. (Steps 322, 324). After decrementing the count, CPU 32 releases the count register 92 by clearing the semaphore 96. (Step 326).

CPU 32 then examines the updated count to determine if the buffer is empty. (Step 328). If the buffer contains another data block, CPU 32 continues to read blocks until the buffer is emptied. (Step 330). Once the buffer is empty, CPU 32 returns to other tasks until a new interrupt appears of bus 62 indicating that new data has been loaded into the data buffer. (Step 332).

The control message blocks 74, 76 operate in essentially the same manner to transfer control messages. More specifically, control register block 72 includes an entry pointer 98 indicating the location of the next available message container, and a removal pointer 100 indicating the location of the next message container to be emptied. (Fig. 4). It further includes a count register 102 indicating the number of control messages stored in the message block 74. Semaphore block 70 includes an associated semaphore 104 for use in arbitrating competing requests for access the count register 102.

Similarly, control register block 72 includes a removal pointer 105, an entry pointer 106, and a count register 107, for the use in controlling the passing of messages through control message block 76. Semaphore block 70 includes a semaphore 108 for arbitrating between competing requests for access to count register 107.

INTERRUPTS

As explained above, once CPU 24 loads the first data or message block to memory 31, (that is memory 31 was previously empty), it notifies CPU 32 that the block is available by interrupting CPU 32. The following describes in more detail the mechanism for generating the interrupt.

Referring to Figs. 2 and 3, interface 30 includes a control status register 66 used by CPU 24 to generate an interrupt to CPU 32. More specifically, CPU 24 writes to register 66 to set certain bits in the register which request interface 30 to interrupt CPU 32. To write to register 66, CPU 24 asserts an address dedicated to the register 66. Decoders 44 and 46 decode the address causing decoder 46 to assert control register access signal R4. Upon receipt of register signal R4, register 66 loads data from data bus DB into its register cells.

CPU 24 requests interface 30 to interrupt CPU 32 by setting interrupt bit IG1 of control status register 66 and loading bits zero through five (MID0-MID5) with the identification code ID1 of CPU 32. The identification code ID1 is applied to a comparator 110 which compares ID1 with an identification code "CODE'" assigned to CPU 32. The comparator output and the interrupt generation bit IG1 are applied to a three input AND gate 112, thereby requesting an interrupt.

Interface 30 includes a second control status register 68, virtually identical to register 66, which is used by CPU 32 to enable and disable interrupt requests from CPU 24 by setting or clearing an interrupt enable bit IE2. The interrupt enable bit IE2 is applied to the third input of AND gate 112. If IE2 is set, the activation of the other two inputs causes the output of AND gate 112 to become asserted, triggering an interrupt latch 114 to set. Once set, interrupt latch 114 asserts an interrupt signal "Interrupt2" on bus 62. CPU 32 clears the interrupt by setting the interrupt pending bit IP2 in its status register 68, thereby causing latch 114 to clear.

CPU 32 interrupts CPU 24 in the same manner. More specifically, it sets interrupt bit IG2 in register 68 and loads bits MID0 - MID5 with the identification code ID2. The bit IG1 is applied to an input of an AND gate 113. The Identification code bits ID2 are applied tɔ an input of a comparator 111. Comparator 111 compares ID2 to "code2". If ID2 and Code2 are identical, the comparator supplies a "match" signal to AND gate 113. Finally, interrupt enable bit IE1 from status register 66 is applied to the third input of AND gate 113. If all three inputs to AND gate 113 are asserted, the output of AND gate 113 sets an interrupt latch 115. Once set, Interrupt latch 115 asserts an interrupt signal "Interrupt1" across cable 19. Interface 28 forwards the interrupt to bus 22 thereby interrupting CPUy24. CPU 24 clears the interrupt by setting interrupt pending bit IP1 in its status register 66.

The interrupt mechanism also provides the means by which the backup fileserver determines when the primary has failed. More specifically, in normal operation, CPU 24 will regularly interrupt CPU 32 with new data to be loaded to the backup disk 34. If CPU 32 does not receive an interrupt within a specified period of time, it assumes the primary has failed and proceeds to assume responsibility for the primary.

It is possible that a properly operating primary will not send any data to the backup fileserver for a long period of time due to a lack of activity on the network. Accordingly, the primary also monitors the length of time since it last interrupted CPU 32. If the specified period of time is about to expire, CPU 24 sends an "Alive" message to control block 74 and interrupts CPU 32 to notify it that the buffer contains a message. CPU 32 will respond to the interrupt, read the Alive message and return to its normal operation. In this manner; CPU 24 notifies CPU 32 that it is operational even during moments when no data needs to be copied to backup disk 34. Similarly, CPU 32 regularly sends alive messages to CPU 24 to notify it that CPU 32 remains operational.

Referring to Figs. 7(a)-7(d), if CPU 24 fails to interrupt CPU 32 for the specified length of time, the backup fileserver assumes responsibility for the primary. The backup fileserver 16 sends a "shut down" message to the block 76 to indicate to the primary that it is taking over. (Step 410). Backup fileserver 16 then mounts the backup filesystem as a local filesystem (Step 412) and activates its network interface 116. (Fig. 1)(Step 416). It then broadcasts an "Address Resolution Protocol" packet (herein "ARP" packet) over link 14 via network interface 116 indicating that it will now handle all traffic formerly directed to the primary. More specifically, each client node of the primary maintains a Node ID table containing the network interface address used by each node recognized by the client's operating system. The "ARP" packet instructs each client node to modify its table by replacing the interface address of network interface 20 in the primary with the address of network interface 116 in the backup. (Step 418).

After sending the ARP packet, the backup fileserver begins maintaining a record of all disk data blocks which are amended so that the primary's outdated data blocks can be replaced at a future time. More specifically, the backup fileserver allocates a block CPU memory for storage of a journal bit map. (Step 420). Each bit in the map corresponds to a single disk data block. If a given block is written to or otherwise modified, the corresponding bit in the journal bit map is set.

To update the newly created journal bit map, the backup fileserver first reads a journal file which describes all changes to the filesystem within the last fifteen seconds. (Step 422). The backup fileserver examines the journal and, for each data block modified in the last fifteen seconds, sets a corresponding bit in the journal bit map. (Step 424). The backup fileserver continues to monitor the journal file, updating the bit map with each modification of the filesystem.

The primary fileserver will eventually be restarted after the failure condition is remedied. Upon being restarted, the primary sends an "Alive" message to control message block 74. The backup fileserver reads the "Alive" message from the control message block and begins returning control to the primary. (Step 426).
Toward this end, it first deactivates its network interface 116 from receiving any more packets. (Step 428). It then waits fifteen seconds for all pending network packets to be processed. (Step 430). More specifically, the backup fileserver may have already received packets from the network which have not yet been incorporated into the filesystem. Further, it may have already begun preparing packets for transmission over the network. Accordingly, during the fifteen second waiting period, the backup fileserver updates the filesystem to reflect any required changes and transmits all pending packets.

After waiting for fifteen seconds, the backup fileserver sends a disk data block allocation bit map to the primary through the dual ported memory 31. (Step 432). Each bit of the block allocation bit map corresponds to a disk data block. A bit set to one indicates that the corresponding disk block is used, a zero indicates that the disk block is free. The backup fileserver then examines the journal bit map to determine if it reflects the most recent changes to the filesystem. (Step 434). If not, the backup fileserver updates the journal bit map to reflect the most recent changes. (Steps 436, 438).

Once the journal bit map is updated, the fileserver scans the journal bit map tc identify each disk data block which has been modified since fifteen seconds prior to the failure of the primary. (Step 440). It then sends each modified data block to the primary through dual ported memory 31 using the data block transfer procedure described above. (Step 442). After completing this transfer, it deallocates the CPU memory block which contains the journal bit map (Step 444) and sends a "Journal Done" message to the primary through control message block 76. (Step 446).

Referring to Figs. 8(a) and 8(b), the following describes the operation of the primary file server 15 in returning to operation from a failure. The primary first sends an "alive" message to the backup fileserver through the control message block 74 of the dual ported memory 31. (Step 510). When the backup fileserver responds by sending the disk data block allocation bit map, the primary replaces its old data block allocation bit map with the newly arrived allocation bit map. (Step 512). It then proceeds to read each arriving modified data block from the dual ported interface and to write the block to its disk 26. (Step 514). When the backup fileserver sends a Journal Done message, indicating that all blocks have been sent (Step 516), the primary activates its network interface (Step 518) and mounts the filesystem (Step 520). It then broadcasts a ARP packet instructing all client nodes to amend their node Id table to indicate that the primary has returned to service (Step 522). It finally sends an "on-line" message to the backup through control message block 76, instructing the backup to return to its role as a backup (Step 524).

When the backup fileserver is first powered on, the contents of the control register block 72 are invalid (i.e., in an unknown state). Accordingly, the semaphore block 70 includes a power up semaphore 109 to provide a mechanism for notifying CPU 24 and CPU 32 that the control entries are not valid. Both CPU 32 and CPU 24 read the semaphore bus when their respective fileserver's are first powered on. Semaphore 109 is initially set to zero when the backup fileserver is powered up to indicate that control register blocký72 is not initialized. The first of CPU 24 and CPU 32 to read a zero in semaphore 109 assumes responsibility for initializing these locations. Upon being read by either CPU, the semaphore is automatically set to a one to indicate to any CPU which subsequently reads the semaphore that the first CPU has assumed responsibility for initialization.

Referring again to Fig.ý1, backup fileserver 16 may operate both as a backup fileserver and as a second primary fileserver. Toward this end, fileserver 16 includes two network interfaces 116, 118. As explained above, interface 116 is used to access communication link 14 when fileserver 16 has assumed responsibility for the failed primary. Interface 118 is used by fileserver 16 to communicate over link 14 in its capacity as a second primary fileserver.

Additions, subtractions, deletions and other modifications of the preferred particular embodiments of the invention will be apparent to those practiced in the art and are within the scope of the following claims.

What is claimed is:

1. In fault tolerant network fileserver system comprising:

a network communication link,

a plurality of nodes connected to the network communication link,

a primary fileserver for storing files from said plurality of nodes connected to the network communication link; and

a backup fileserver for storing copies of files from said primary fileserver;

the improvement comprising

said primary fileserver comprising:

a primary computer processor,

a primary storage disk,

a first network interface connected to said network communication link, and

a first independent interface connected to said backup fileserver and responsive to commands from the primary processor for communicating information to said backup fileserver;

said backup fileserver comprising:

a backup computer processor,

a backup storage disk,

a backup network interface connected to said network communication link, and

a second independent interface connected to said first independent interface for receiving information from said primary fileserver and responsive to commands from said back-up processor for reading information stored in said memory, said second independent interface comprising a dual ported memory for storing information received from said primary fileserver.

2. The fileserver system of claim 1 wherein said second independent interface further comprises an interrupt means for interrupting said backup computer processor to notify said backup computer processor that said dual ported memory contains information received from said primary fileserver.

3. The fileserver system of claim 1 wherein said dual ported memory comprises:

a group of data memory blocks each memory block for storing data to be copied to said backup storage disk,

an entry pointer register for storing an entry pointer identifying a next available data block,

a removal pointer register for storing a removal pointer identifying a next data block to be emptied, and

a count register for storing the number of data memory blocks containing data.

4. The fileserver system of claim 3 wherein said dual ported memory further comprises:

a first control message block for storing control messages from said primary fileserver to said backup fileserver, and

a second control message block for storing control messages from said backup fileserver to said primary fileserver.

5. The fileserver system of claim 3 wherein said second independent interface further comprises means responsive to signals from said first independent interface for modifying the contents of a common register assessable to the backup computer processor to indicate the entry of data into a memory block, and wherein said second independent interface is further

responsive to signals from the backup computer processor
for modifying said common register to indicate the removal of data from a memory block, said dual ported memory comprising a semaphore register for arbitrating access to said common register.

6. The fileserver system of claim 5 wherein said semaphore register has a first and a second state of operation; whereby when read by a processor while in said first state, the semaphore register provides a register available code and automatically enters said second state; when read by a processor while in said second state, said semaphore register provides a register unavailable code and remains in said second state; and when written to by a processor while in said second state, said semaphore register returns to said first state.

7. The fileserver system of claim 6 wherein said common register is said count register, and wherein said primary fileserver has means for incrementing said count register to indicate the entry of data in a data memory block and said backup processor has means for decrementing said count register to indicate the removal of data from a data memory block.

8. The fileserver system of claim 1 wherein said dual ported memory comprises at least one semaphore bit having a first and a second state of operation; whereby said at least one semaphore bit is in said first state when said second independent interface is powered on; and said at least one semaphore bit automatically enters said second state when read by a processor.

9. The fileserver system of claim 1 wherein said primary fileserver further comprises:

a disk operating system for controlling the copying of files received by said first network interface to said primary storage disk and to said dual ported memory, said disk operating system including means for receiving secure disk write instructions and unsecure disk write instructions directing said operating system to write specified files to said primary storage disk, said secure write instruction requesting a write to said primary storage disk more promptly than said unsecure write instruction,

said operating system responding to both secure and unsecure instructions by promptly writing said specified files to said dual ported memory, and by writing said specified files to said primary disk storage device using an unsecure write procedure.

10. The fileserver system of claim 8 wherein said disk operating system is a modified Unix operating system which, in response to synchronous disk write instructions, promptly writes said specified files to said dual ported memory, and writes said specified files to said primary storage disk using a delayed write procedure.

11. A method for maintaining on a backup network node, a backup filesystem which mirrors a primary filesystem of a primary network node, the method comprising the steps of:

receiving, at said primary node, secure disk write instructions directing said primary node to write other specified files to a primary nonvolatile storage device using a secure write procedure;

receiving, at said primary node, unsecure disk write instructions directing said primary node to write specified files to a primary nonvolatile storage device using an unsecure write procedure, said secure write procedure writing said specified files to said primary nonvolatile storage device more promptly than said unsecure write procedure; and

in response to either of said secure and unsecure instructions,

copying said specified files to a shared memory within said backup node, and

writing said specified files to said nonvolatile storage device using said unsecure write procedure.

12. The method of claim 11 wherein writing a specified file to said shared memory comprises the steps of:

reading from an entry pointer register within said shared memory, an entry pointer identifying a next available data block portion of said shared memory, and

writing said specified file to said next available data block; and wherein said method further comprises the steps of:

reading from a removal pointer register within said shared memory, a removal pointer identifying a next data block portion of said shared memcry to be emptied, and

copying said next data block portion to a storage device for storing said backup filesystem.

13. The method of claim 12 wherein writing a specified file to said shared memory further comprises the steps of:

reading from a count register within said shared memory, a count identifying the number of data blocks within said data block portion of said shared memory which contain specified files, and

incrementing said count register to indicate said writing of a specified file to a next available data block; and wherein said method further comprises the step of:

decrementing said count register to indicate said reading of a specified file from a next data block portion to be emptied.

14. The method of claim 13 wherein said method further comprises the steps of:

reading from a semaphore register within said shared memory, an available code indicating whether said count register is available, and

resetting said semaphore register after said incrementing or decrementing of said count register to indicate that said count register is available.

15. The method of claim 12 further comprising the steps of:

reading from a power up semaphore register within said shared memory, an initialization code indicating whether selected registers within said shared memory have been initialized since said shared memory was powered on, and

initializing said selected registers if said initialization code indicates that said selected registers have not been initialized.

16. The method of claim 11 further comprising the step of:

interrupting a backup processor within said backup node to notify said backup node that said primary node is operational.

17. The method of claim 16 wherein interrupting said backup processor comprises the steps of:

writing an alive control message to a control message block within said shared memory, and interrupting said backup processor to notify said processor that said shared memory includes a control message.

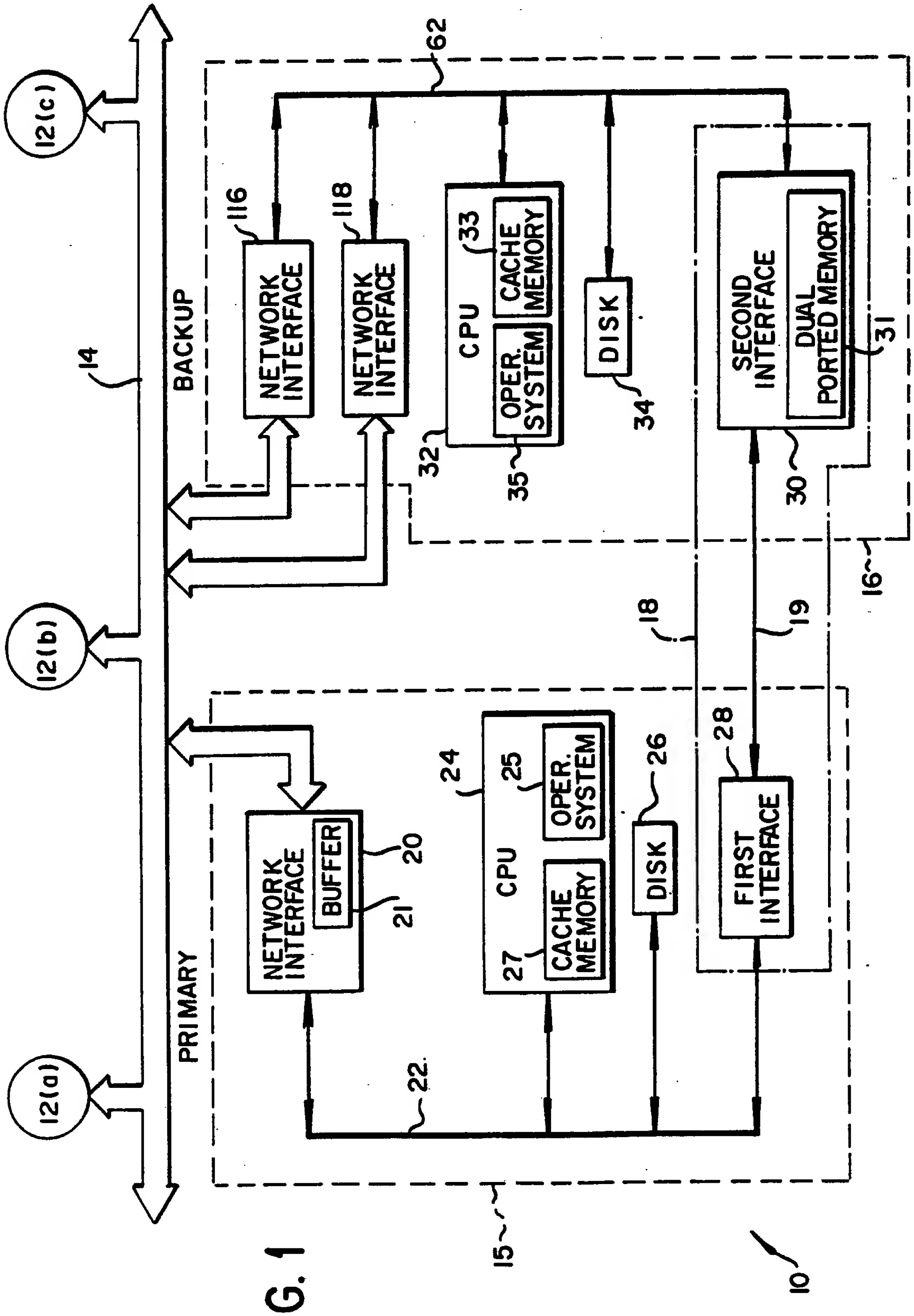

12(a)
12(b)
12(c)
14
PRIMARY
BACKUP
NETWORK INTERFACE
BUFFER
20
21
22
CPU
24
25
OPER. SYSTEM
CACHE MEMORY
27
DISK
26
FIRST INTERFACE
28
15
10
18
19
16
62
116
118
NETWORK INTERFACE
NETWORK INTERFACE
CPU
33
CACHE MEMORY
OPER. SYSTEM
32
35
DISK
34
SECOND INTERFACE
DUAL PORTED MEMORY
31
30

FIG. 1

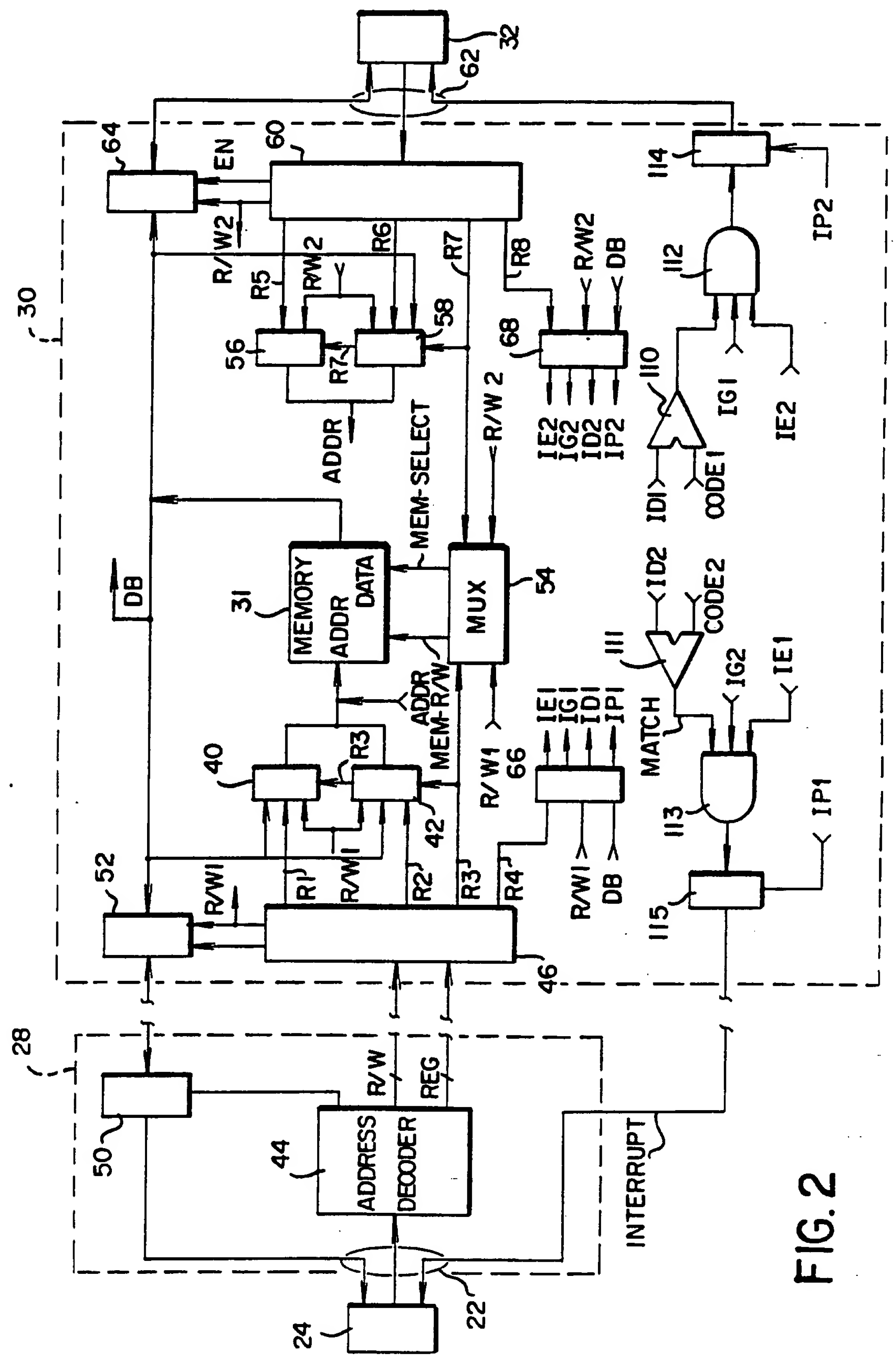
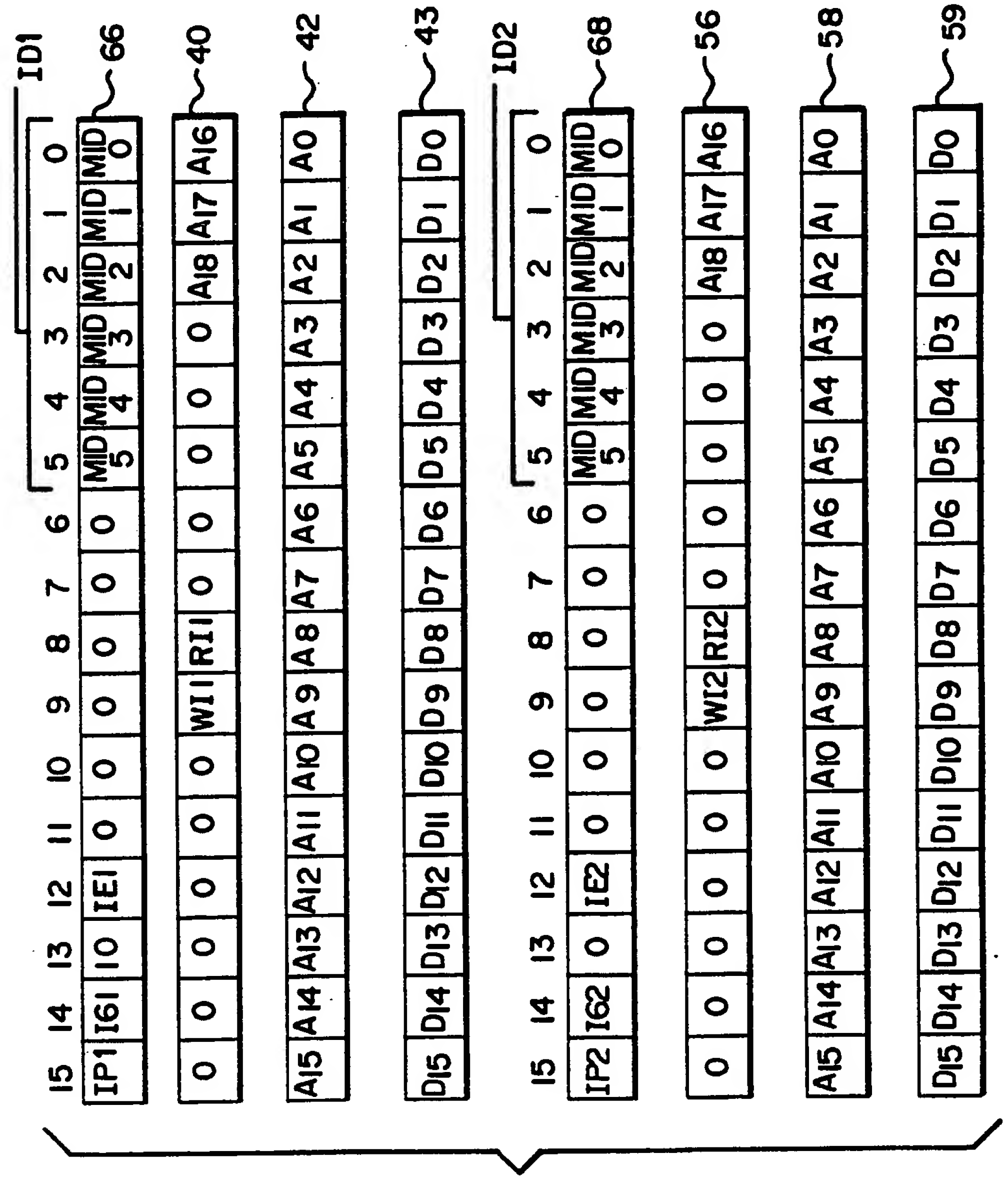

32
62
30
64
EN
60
R/W2
R5
R/W2
R6
R7
R8
R/W2
DB
114
IP2
112
56
R7
58
68
IE2
IG2
ID2
IP2
110
IG1
IE2
ADDR
MEM-SELECT
R/W 2
ID1
CODE1
DB
31
MEMORY
ADDR
DATA
MUX
54
ID2
CODE2
111
IG2
IE1
MATCH
ADDR
MEM-R/W
40
R3
42
R/W1
66
IE1
IG1
ID1
IP1
113
IP1
52
R/W1
R1
R/W1
R2
R3
R4
R/W1
DB
115
46
28
R/W
REG
50
44
ADDRESS
DECODER
INTERRUPT
24
22

FIG. 2

FIG. 3



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
66	IP1	I61	IO	IE1	0	0	0	0	0	0	MID 5	MID 4	MID 3	MID 2	MID 1	MID 0
40	0	0	0	0	0	0	WI1	RI1	0	0	0	0	0	A18	A17	A16
42	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
43	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
68	IP2	I62	0	IE2	0	0	0	0	0	0	MID 5	MID 4	MID 3	MID 2	MID 1	MID 0
56	0	0	0	0	0	0	WI2	RI2	0	0	0	0	0	A18	A17	A16
58	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
59	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

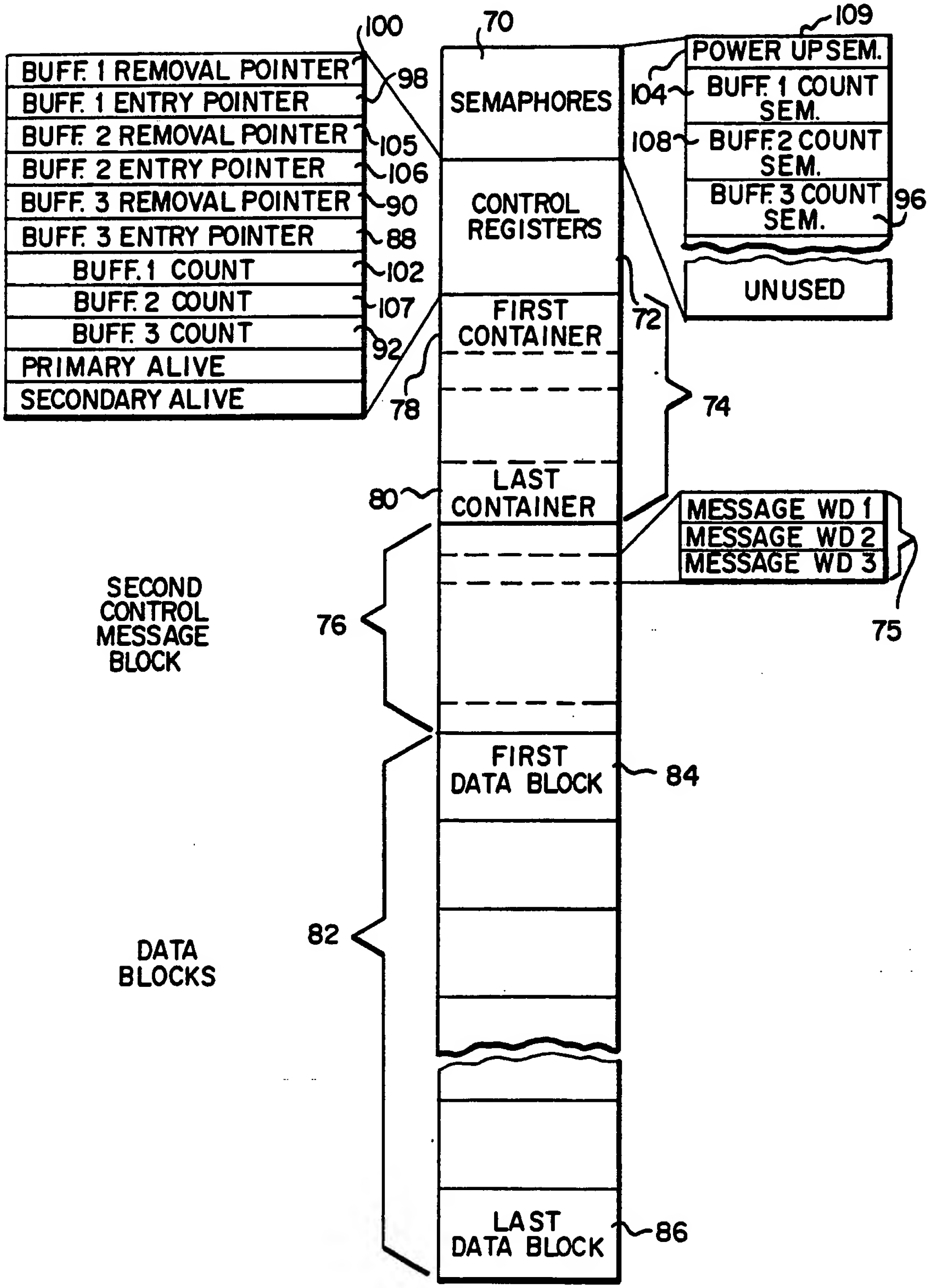

100
70
109
BUFF. 1 REMOVAL POINTER
BUFF. 1 ENTRY POINTER
BUFF. 2 REMOVAL POINTER
BUFF. 2 ENTRY POINTER
BUFF. 3 REMOVAL POINTER
BUFF. 3 ENTRY POINTER
BUFF. 1 COUNT
BUFF. 2 COUNT
BUFF. 3 COUNT
PRIMARY ALIVE
SECONDARY ALIVE
98
105
106
90
88
102
107
92
78
SEMAPHORES
CONTROL REGISTERS
FIRST CONTAINER
LAST CONTAINER
80
72
74
POWER UP SEM.
BUFF. 1 COUNT SEM.
BUFF. 2 COUNT SEM.
BUFF. 3 COUNT SEM.
UNUSED
104
108
96
MESSAGE WD 1
MESSAGE WD 2
MESSAGE WD 3
75
SECOND CONTROL MESSAGE BLOCK
76
FIRST DATA BLOCK
84
DATA BLOCKS
82
LAST DATA BLOCK
86

FIG. 4

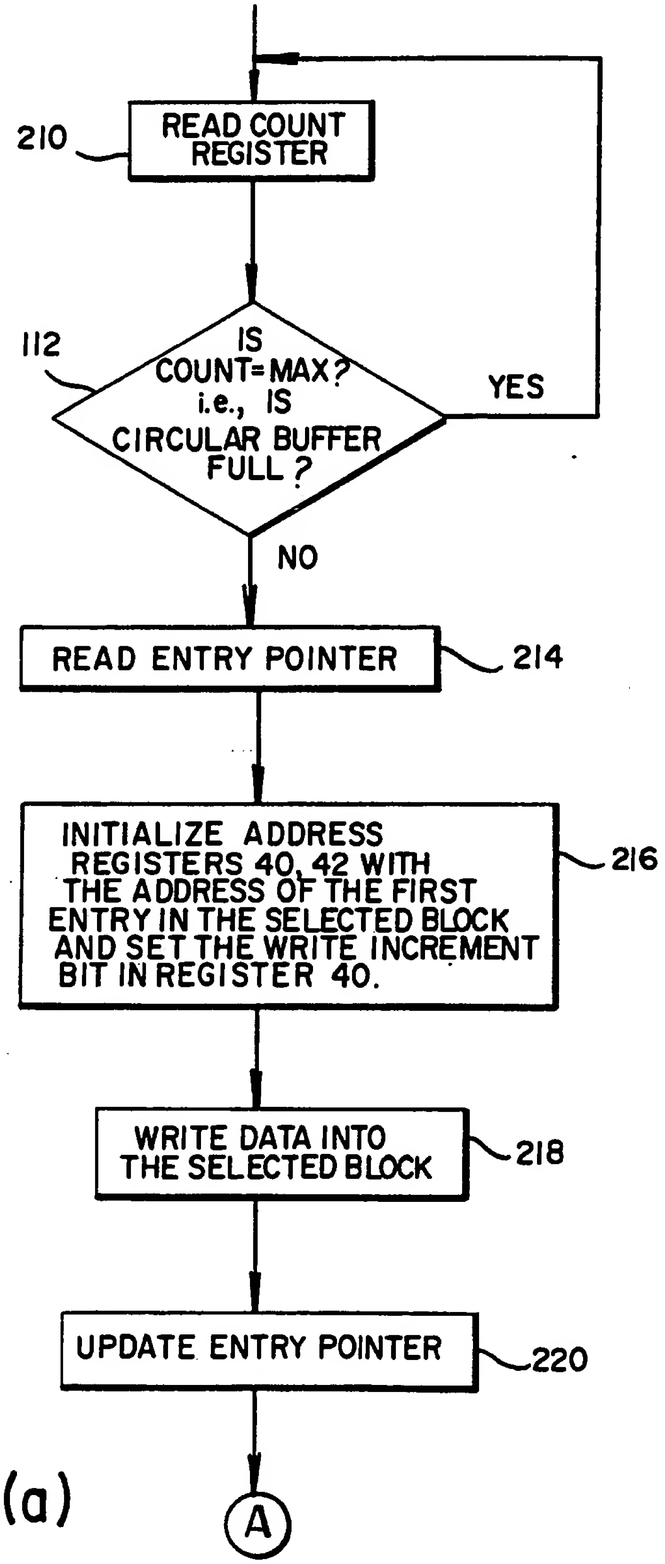

READ COUNT REGISTER
210
IS COUNT=MAX? i.e., IS CIRCULAR BUFFER FULL?
112
YES
NO
READ ENTRY POINTER
214
INITIALIZE ADDRESS REGISTERS 40, 42 WITH THE ADDRESS OF THE FIRST ENTRY IN THE SELECTED BLOCK AND SET THE WRITE INCREMENT BIT IN REGISTER 40.
216
WRITE DATA INTO THE SELECTED BLOCK
218
UPDATE ENTRY POINTER
220
A

FIG. 5(a)

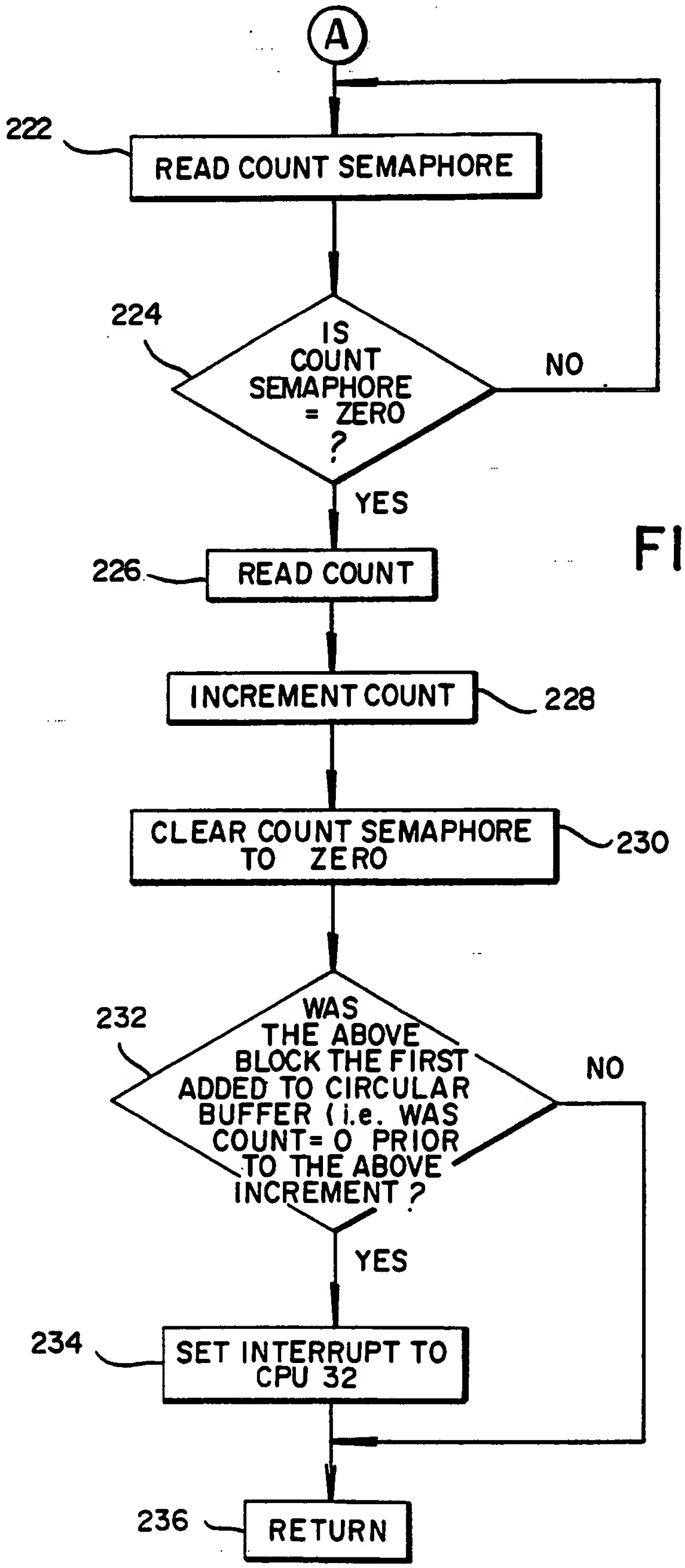

A
222
READ COUNT SEMAPHORE
224
IS COUNT SEMAPHORE = ZERO ?
NO
YES
226
READ COUNT
INCREMENT COUNT
228
CLEAR COUNT SEMAPHORE TO ZERO
230
232
WAS THE ABOVE BLOCK THE FIRST ADDED TO CIRCULAR BUFFER (i.e. WAS COUNT= 0 PRIOR TO THE ABOVE INCREMENT ?
NO
YES
234
SET INTERRUPT TO CPU 32
236
RETURN

FIG. 5(b)

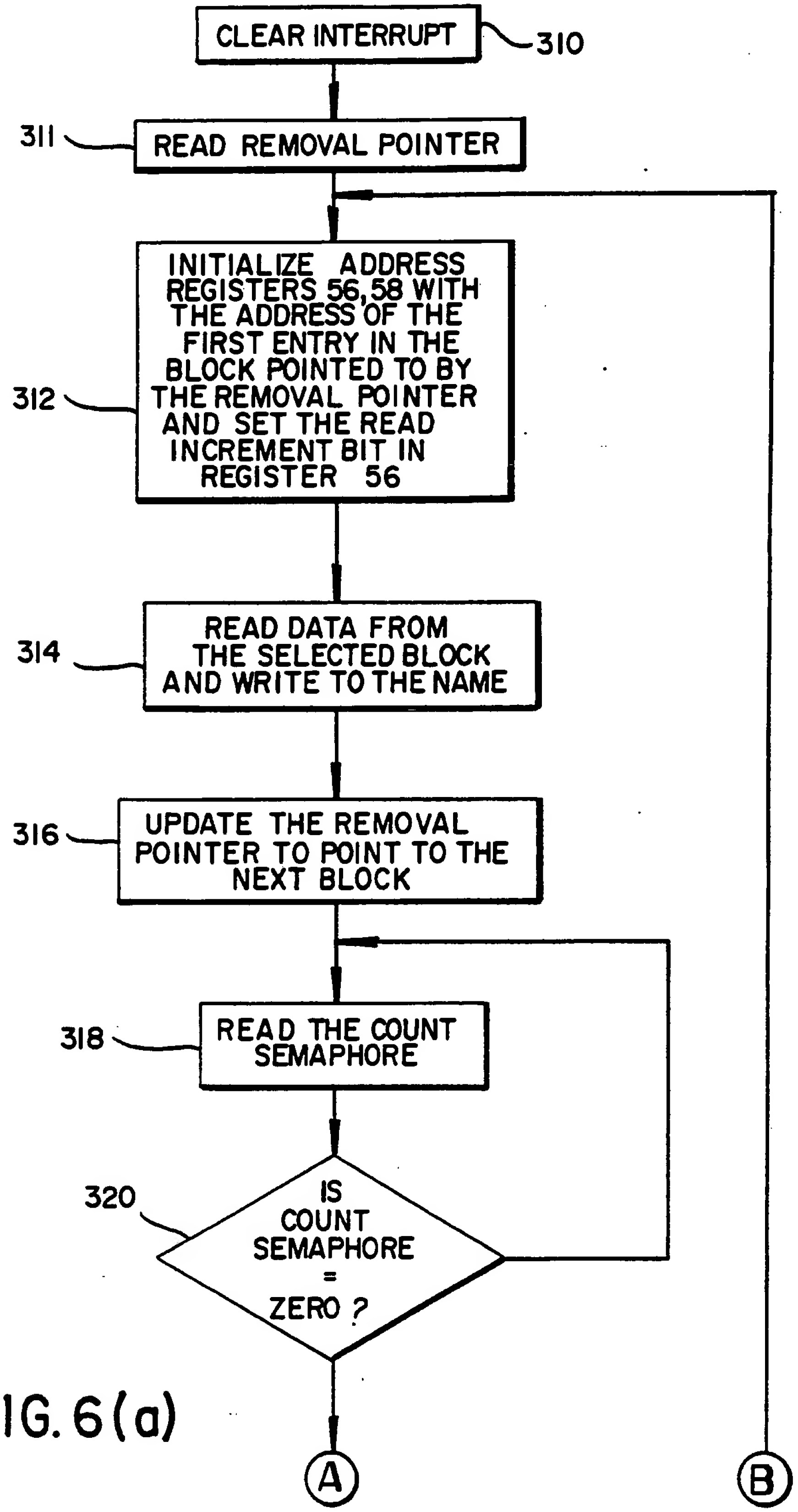

CLEAR INTERRUPT
310
311
READ REMOVAL POINTER
INITIALIZE ADDRESS REGISTERS 56,58 WITH THE ADDRESS OF THE FIRST ENTRY IN THE BLOCK POINTED TO BY THE REMOVAL POINTER AND SET THE READ INCREMENT BIT IN REGISTER 56
312
314
READ DATA FROM THE SELECTED BLOCK AND WRITE TO THE NAME
316
UPDATE THE REMOVAL POINTER TO POINT TO THE NEXT BLOCK
318
READ THE COUNT SEMAPHORE
320
IS COUNT SEMAPHORE = ZERO ?
A
B

FIG. 6(a)

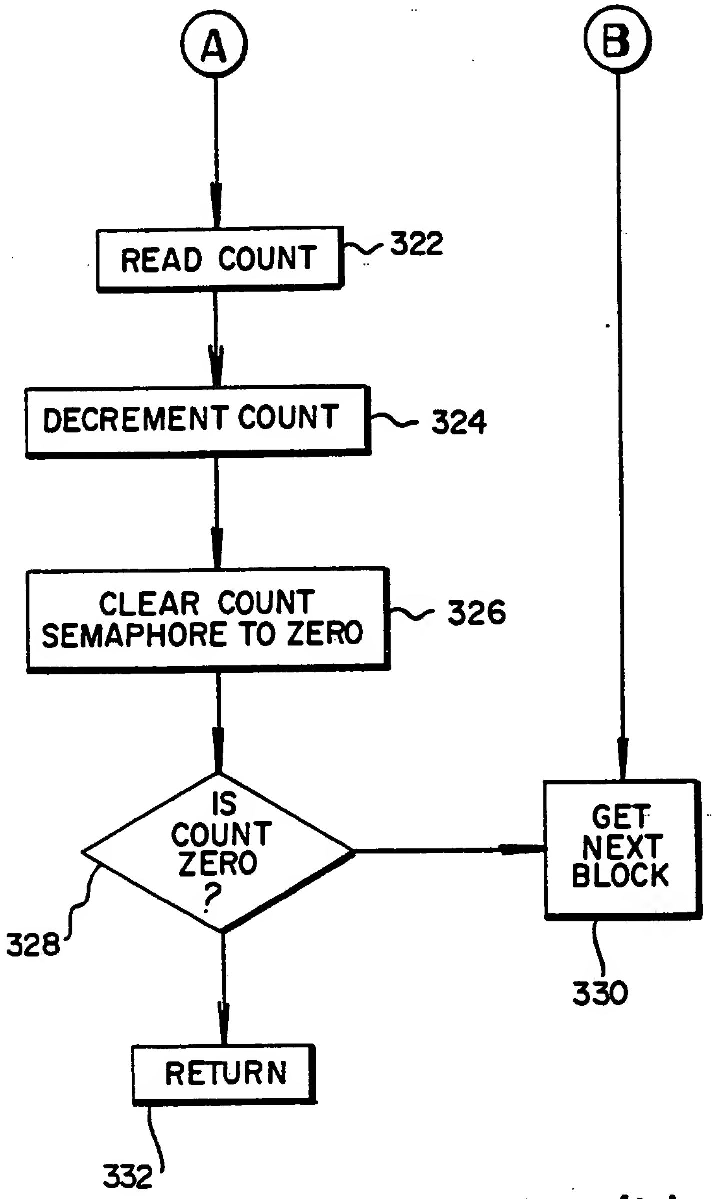

A
B
READ COUNT
322
DECREMENT COUNT
324
CLEAR COUNT SEMAPHORE TO ZERO
326
IS COUNT ZERO ?
328
GET NEXT BLOCK
330
RETURN
332

FIG. 6(b)

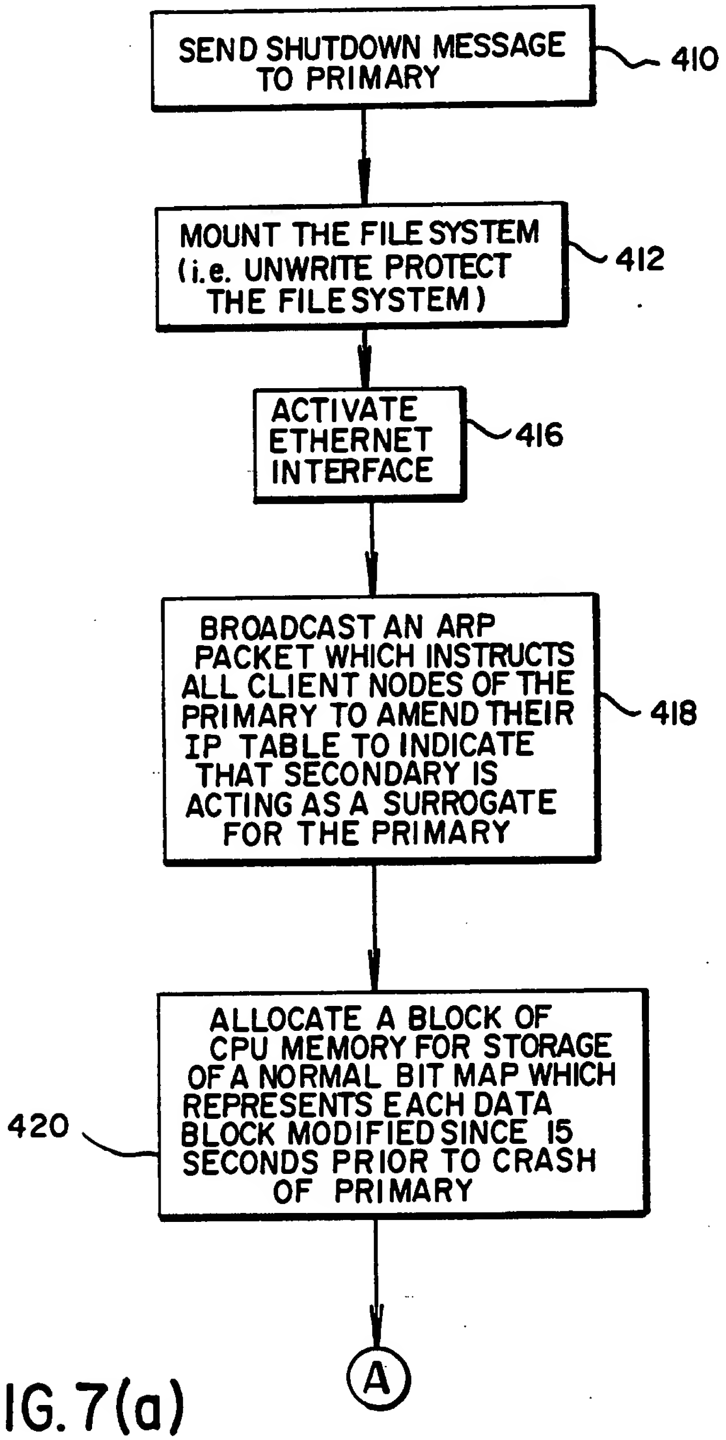

SEND SHUTDOWN MESSAGE TO PRIMARY
410
MOUNT THE FILE SYSTEM (i.e. UNWRITE PROTECT THE FILE SYSTEM)
412
ACTIVATE ETHERNET INTERFACE
416
BROADCAST AN ARP PACKET WHICH INSTRUCTS ALL CLIENT NODES OF THE PRIMARY TO AMEND THEIR IP TABLE TO INDICATE THAT SECONDARY IS ACTING AS A SURROGATE FOR THE PRIMARY
418
ALLOCATE A BLOCK OF CPU MEMORY FOR STORAGE OF A NORMAL BIT MAP WHICH REPRESENTS EACH DATA BLOCK MODIFIED SINCE 15 SECONDS PRIOR TO CRASH OF PRIMARY
420
A

FIG. 7(a)

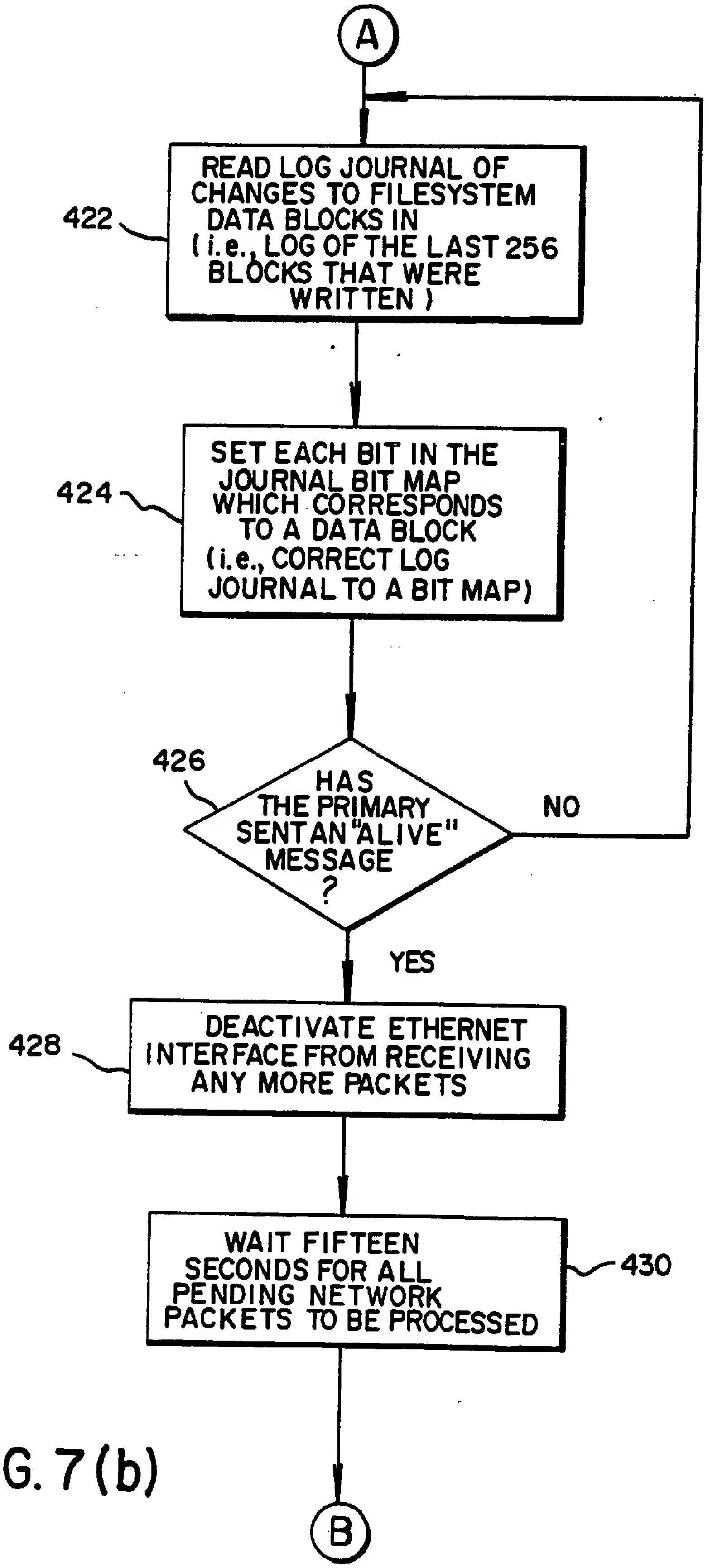

A
READ LOG JOURNAL OF CHANGES TO FILESYSTEM DATA BLOCKS IN (i.e., LOG OF THE LAST 256 BLOCKS THAT WERE WRITTEN)
422
SET EACH BIT IN THE JOURNAL BIT MAP WHICH CORRESPONDS TO A DATA BLOCK (i.e., CORRECT LOG JOURNAL TO A BIT MAP)
424
HAS THE PRIMARY SENT AN "ALIVE" MESSAGE ?
426
NO
YES
DEACTIVATE ETHERNET INTERFACE FROM RECEIVING ANY MORE PACKETS
428
WAIT FIFTEEN SECONDS FOR ALL PENDING NETWORK PACKETS TO BE PROCESSED
430
B

FIG. 7(b)

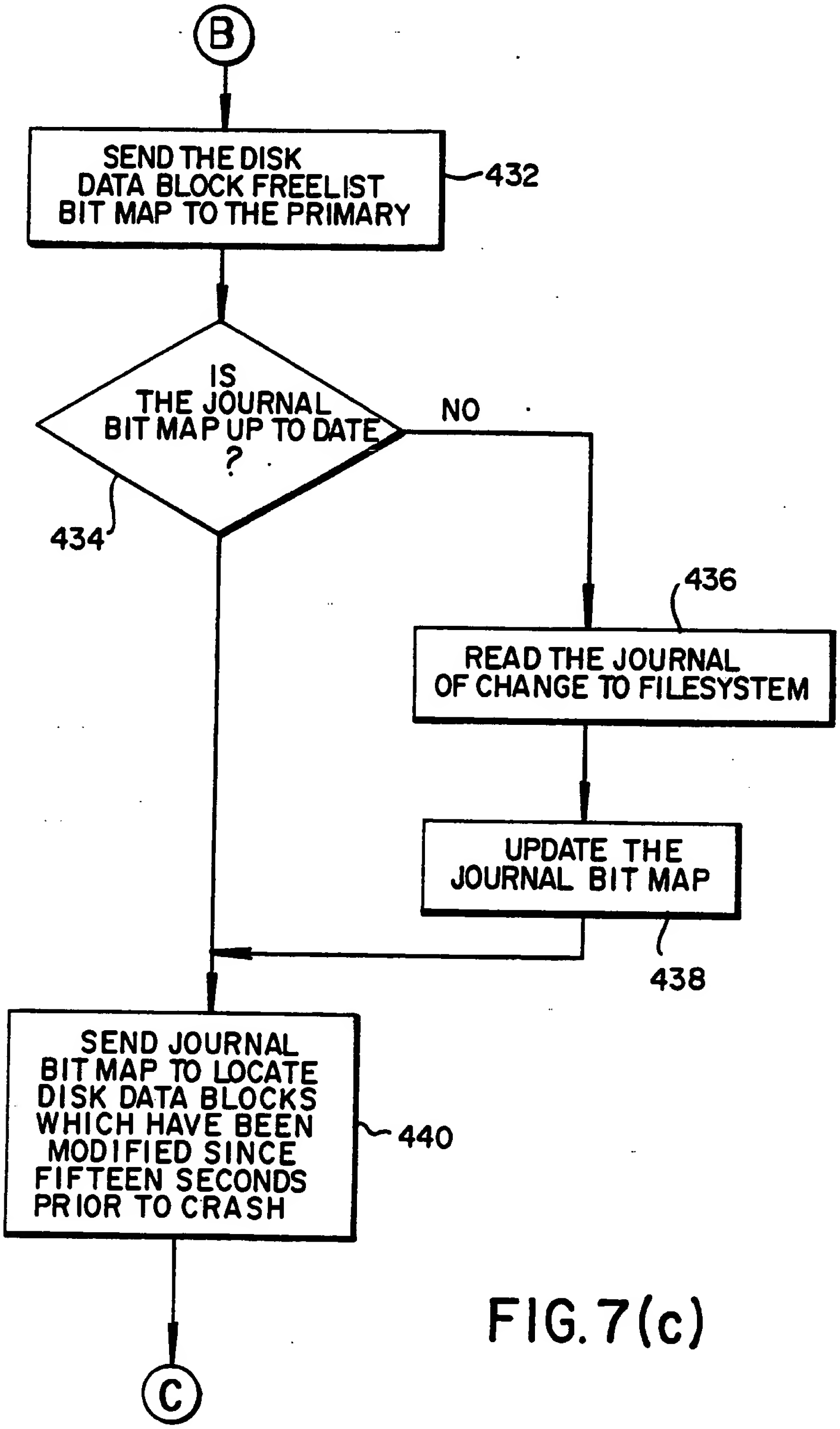

B
SEND THE DISK DATA BLOCK FREELIST BIT MAP TO THE PRIMARY
432
IS THE JOURNAL BIT MAP UP TO DATE ?
NO
434
436
READ THE JOURNAL OF CHANGE TO FILESYSTEM
UPDATE THE JOURNAL BIT MAP
438
SEND JOURNAL BIT MAP TO LOCATE DISK DATA BLOCKS WHICH HAVE BEEN MODIFIED SINCE FIFTEEN SECONDS PRIOR TO CRASH
440
C

FIG. 7(c)

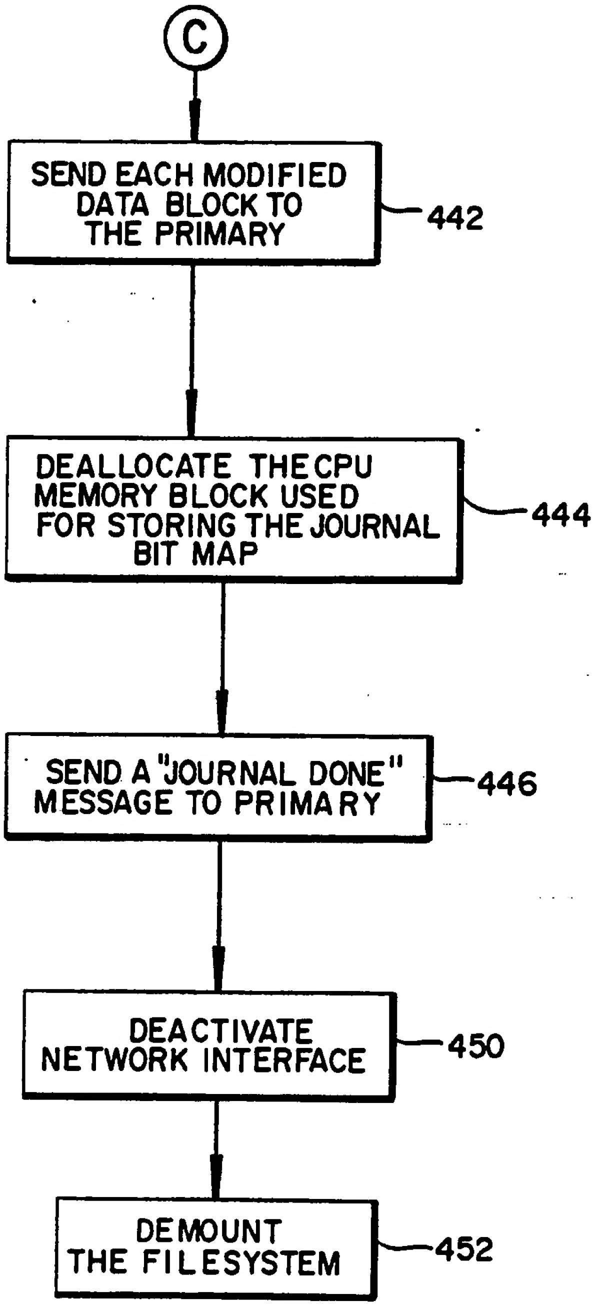

C
SEND EACH MODIFIED DATA BLOCK TO THE PRIMARY
442
DEALLOCATE THE CPU MEMORY BLOCK USED FOR STORING THE JOURNAL BIT MAP
444
SEND A "JOURNAL DONE" MESSAGE TO PRIMARY
446
DEACTIVATE NETWORK INTERFACE
450
DEMOUNT THE FILESYSTEM
452

FIG. 7(d)

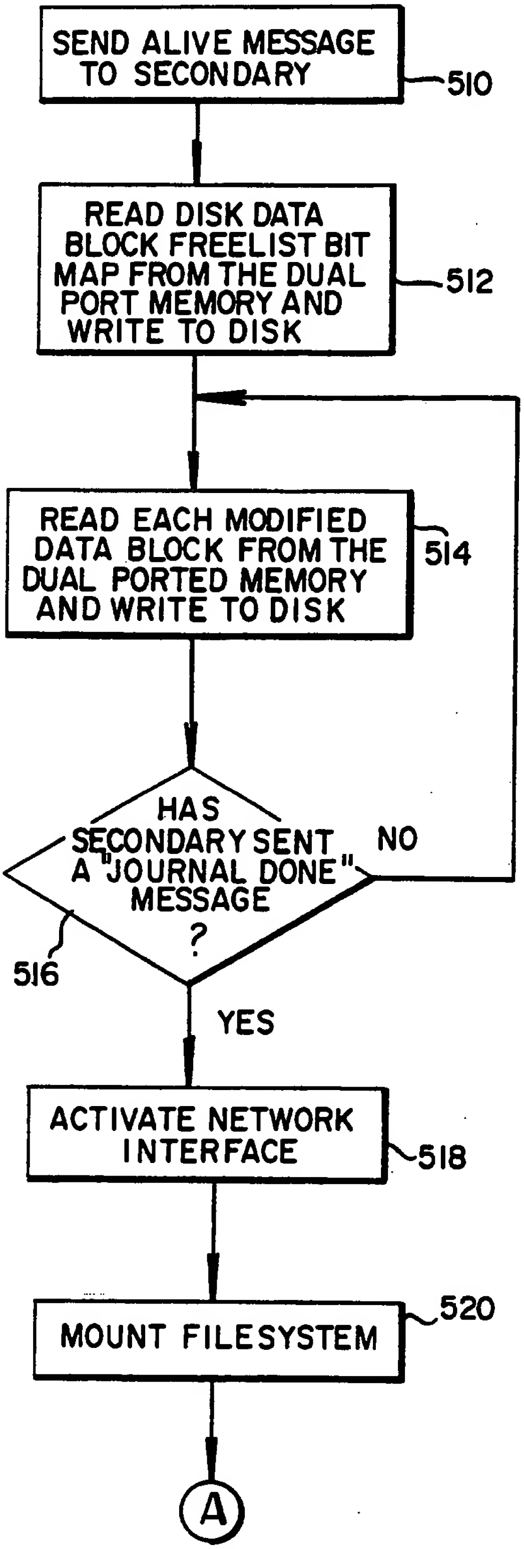

SEND ALIVE MESSAGE TO SECONDARY
510
READ DISK DATA BLOCK FREELIST BIT MAP FROM THE DUAL PORT MEMORY AND WRITE TO DISK
512
READ EACH MODIFIED DATA BLOCK FROM THE DUAL PORTED MEMORY AND WRITE TO DISK
514
HAS SECONDARY SENT A "JOURNAL DONE" MESSAGE ?
NO
516
YES
ACTIVATE NETWORK INTERFACE
518
MOUNT FILESYSTEM
520
A

FIG. 8(a)

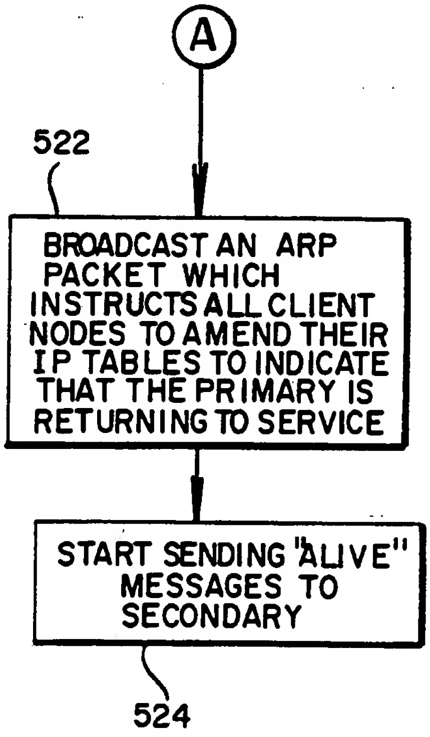

A
522
BROADCAST AN ARP PACKET WHICH INSTRUCTS ALL CLIENT NODES TO AMEND THEIR IP TABLES TO INDICATE THAT THE PRIMARY IS RETURNING TO SERVICE
START SENDING "ALIVE" MESSAGES TO SECONDARY
524

FIG. 8(b)

INTERNATIONAL SEARCH REPORT

International Application No PCT/US 92/03001

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all)[6]

According to International Patent Classification (IPC) or to both National Classification and IPC

Int.Cl. 5 G06F11/20; G06F11/14

II. FIELDS SEARCHED

Minimum Documentation Searched[7]

Classification System	Classification Symbols
Int.Cl. 5	G06F

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched[8]

III. DOCUMENTS CONSIDERED TO BE RELEVANT[9]

Category°	Citation of Document,[11] with indication, where appropriate, of the relevant passages[12]	Relevant to Claim No.[13]
A	US,A,4 958 270 (P. F. MACLAUGHLIN, P. H. MODY) 18 September 1990 see column 2, line 54 - column 5, line 28 see column 6, line 66 - column 7, line 10 see figures 1,3,4 ---	1,3,4,9, 11
A	EP,A,0 359 471 (COMPAQ COMPUTER CORP.) 21 March 1990 see column 1, line 10 - column 3, line 15 see figures 1-3 ---	1,9,11
A	WO,A,8 909 452 (NCR CORP.) 5 October 1989 see page 4, line 19 - page 5, line 14 see page 6, line 22 - page 7 see page 8, line 17 - page 10 see figures 1-4 --- -/--	1,3,5,11

° Special categories of cited documents :[10]

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search 13 JULY 1992	Date of Mailing of this International Search Report 20 JUIL. 1992
International Searching Authority EUROPEAN PATENT OFFICE	Signature of Authorized Officer JOHANSSON U.C.

1

International Application No PCT/US 92/03001

III. DOCUMENTS CONSIDERED TO BE RELEVANT (CONTINUED FROM THE SECOND SHEET)

Category °	Citation of Document, with indication, where appropriate, of the relevant passages	Relevant to Claim No.
A	ACM TRANSACTIONS ON COMPUTER SYSTEMS vol. 7, no. 1, February 1989, NEW YORK US pages 1 - 24; A. BORG ET AL: 'Fault-Tolerance Under UNIX' see page 3, line 26 - line 44 see page 12, line 21 - page 14, line 8 ---	1,4,9, 10,11

ANNEX TO THE INTERNATIONAL SEARCH REPORT ON INTERNATIONAL PATENT APPLICATION NO. US 9203001 SA 59094

This annex lists the patent family members relating to the patent documents cited in the above-mentioned international search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information. 13/07/92

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
US-A-4958270	18-09-90	EP-A-	0460308	11-12-91
EP-A-0359471	21-03-90	JP-A-	2135550	24-05-90
WO-A-8909452	05-10-89	JP-A- EP-A-	1253061 0377684	09-10-89 18-07-90

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

AM 1

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 767 427 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
09.04.1997 Bulletin 1997/15

(51) Int. Cl.^{6}: **G06F 9/46**, G06F 15/16

(21) Application number: 96203153.0

(22) Date of filing: 13.09.1989

(84) Designated Contracting States:
AT BE CH DE FR GB IT LI LU NL SE

(30) Priority: 13.09.1988 US 244691
13.09.1988 US 244503
13.09.1988 US 244834
13.09.1988 US 244742
13.09.1988 US 244114
13.09.1988 US 244845
13.09.1988 US 244919
13.09.1988 US 244851
13.09.1988 US 244730
13.09.1988 US 244850
13.09.1988 US 244495
07.09.1989 US 402391

(62) Document number(s) of the earlier application(s) in accordance with Art. 76 EPC:
89910805.4 / 0 441 798

(71) Applicant: DIGITAL EQUIPMENT CORPORATION
Maynard Massachusetts 01754-1418 (US)

(72) Inventors:
- Rogers, Dennis
  Leominster, MA 01453 (US)
- Smith, Danny L.
  Haverhill, MA 01830 (US)
- O'Brien, Linsey B.
  Wellesley, MA 02181 (US)
- Ross, Robert R.N.
  Mansfield, MA 02048 (US)
- Schuchard, Robert C.
  Ashburnham, MA 01430 (US)
- Chan-Lizardo, Christine C.
  Chelmsford, MA 01824 (US)
- Callander, Jill F.
  Hudson, MA 01749 (US)
- Goldfarb, Stanley I.
  Hudson, MA 01749 (US)
- Fehskens, Leonard G.
  403E, Westboro, MA 01581 (US)
- Rosenbaum, Richard L.
  Pepperell, MA 01363 (US)
- Namoglu, Sheryl E.
  Mont Vernon, NH 03052 (US)
- Sylor, Mark W.
  Nashua, NH 03062 (US)
- Seger, Mark J.
  Harvard, MA 02451 (US)
- Lemmon, James L., Jr.
  Leominster, MA 02453 (US)
- Shurtleff, David L.
  Boxborough, MA 01719 (US)
- Strutt, Colin
  Westford, MA 01886 (US)
- Trasatti, Philip J.
  Brokkline, NH 03033 (US)
- Adams, William C., Jr.
  Topsfield, MA 01983 (US)
- Dixon, Timothy M.
  Woodcote, Reading RG8 0QD (GB)
- Koning, G. Paul
  Brookline, NH 03033 (US)
- Chapman, Kenneth W.
  Nashua, NH 03063 (US)
- Nelson, Kathy Jo
  Nashua, NH 03062 (US)
- Fletcher, Douglas R.
  Lunenburg, MA 01462 (US)
- Kohls, Ruth E.J.
  Acton, MA 01720 (US)
- Wong, Steven K.
  Chelmsford, MA 01824 (US)
- Dang, Reena
  Lexington, MA 02173 (US)
- Moore, Allan B.
  Acton, MA 01720 (US)
- Navkal, Anil V.
  Maynard, MA 01754 (US)
- England, Benjamin M.
  Haverhill, MA 01831 (US)
- Sankar, Arundahati G.
  Andover, MA 01810 (US)
- Plouffe, Gerard R.
  Nashua, NH 03063 (US)
- Roberts, D. Keith
  Pepperell, MA 01463 (US)
- Guertin, Matthew W.
  Westford, MA 01886 (US)
- Koch, Pamela J.
  Hudson, NH 03051 (US)
- Burgess, Peter H.
  Salisbury, MA 01952 (US)
- Rosenberg, Jeff
  Leominster, MA 01453 (US)

Printed by Rank Xerox (UK) Business Services
2.14.1/3.4

(Cont. next page)

- **Densmore, Michael**
  Chelmsford, MA 01824 (US)
- **Hupper, Theodore F.**
  Marlborough, MA 01752 (US)
- **Aronson, David**
  Boston, MA 02131 (US)
- **Zolfonoon, Riaz**
  Nashua, NH 03062 (US)

(74) Representative: **Dubois-Chabert, Guy et al**
Société de Protection des Inventions
25, rue de Ponthieu
75008 Paris (FR)

Remarks:
This application was filed on 08 - 11 - 1996 as a divisional application to the application mentioned under INID code 62.

(54) Entity management system

(57) A system for managing an assemblage of entities. The entities interface within the assemblage for control of primary information handling functions and further interface with the system to permit the carrying out of management functions. The system includes management modules adapted to carry out management functions by independently interpreting and executing commands, a kernel including a table of dispatch pointers for directing the commands to the respective modules in which they are to be interpreted and executed, and an enroller for enrolling new modules into the system by adding further pointers to the table. In addition, the system includes: a module adapted to independently interpret and execute selected management-related commands; stored records relating to accessed management information, each record indicating an associated time; an information manager, responsive to commands having a time schedule, for retrieving information from the records or accessing information from the entities, including a scheduler for issuing subsidiary accesses or retrievals at possibly multiple times according to the schedule; storage containing domain information defining groups of entities, where the kernel may issue a commands to a group by issuing individual commands to appropriate modules; a common command syntax including fields for identifying the entity and the operation to be performed; a module that stores rules identifying alarm conditions, including a generator for generating rules and an alarm detector for detecting an alarm condition in response to the rules; a module adapted to carry out self-management functions by interpreting and executing commands.

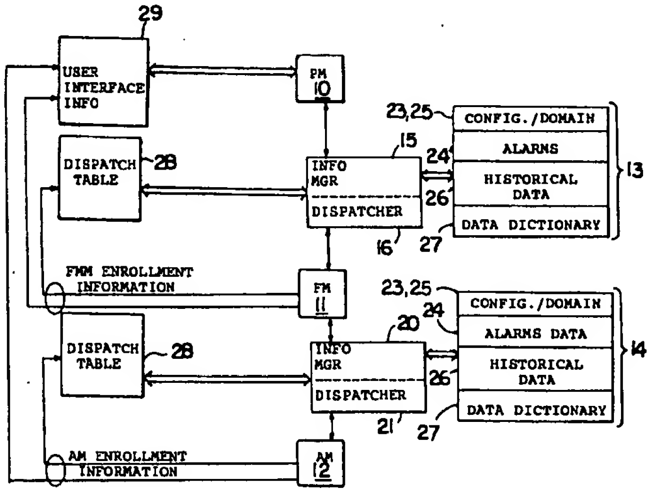


FIG. 5

Description

Background of the Invention

The invention relates generally to the field of management of complex systems, and more particularly to arrangements for managing complex systems such as distributed digital data processing systems.

As digital data processing systems, or computers, have become smaller and less expensive, individual computers are being used by individuals and small groups. To enhance sharing of data, communications among users and economy in connection with resources which may be infrequently used by an individual, computers have been connected into networks which communicate by means of messages transmitted over communications links, which include, in addition to the computers used directly by the various users, servers which, for example, store large amounts of data which may be accessed, used and updated by a number of users in the system, thereby facilitating sharing of data. Servers may also control printers, telecommunications links, and so forth. In addition, servers may provide specialized computational services, such as database searching and sorting, and so forth. The various computers, which are termed clients, and servers are interconnected by a communications link to permit messages to be transferred among the various computers and servers comprising the distributed system.

Summary of the Invention

The invention provides a new and improved control arrangement for controlling and monitoring a complex system, such as a distributed digital data processing system in which a plurality of computers communicate over, for example, a local area network.

In brief summary, the control arrangement includes one or more presentation modules, functional modules and access modules that communicate through kernel means to process reguests generated in response to commands from an operator, and to display responses to the operator. The presentation modules handle operator interface functions, including receipt of commands from an operator and presentation of responses thereto. In response to a command from an operator, the presentation module generates a request. The kernel means receives a request and may route it to a functional module for further processing. The functional modules handle general functional operations in connection with processing a request. In response to a request, a functional module generates one or more requests (sometimes for convenience called subsidiary requests in what follows) that it transfers to the kernel means or to other functional modules for processing. The kernel means routes subsidiary requests which it receives to an access module for processing. The access modules handle primitive operations in connection with the entities comprising the complex system.

In general, in one aspect, the invention features a system for controlling and carrying out management functions over an assemblage of entities, wherein the entities interface within the assemblage for control of primary information handling functions and the entities further interface with the system to permit the carrying out of the management functions. The system includes stored management modules adapted to carry out the management functions by independently interpreting and executing selected management-related commands, a kernel comprising a table of dispatch pointers for directing the commands to the respective modules in which they are to be interpreted and executed, and an enroller for enrolling new management modules into the system by adding further pointers to the table.

Preferred embodiments of this aspect include the following features. The management modules are adapted for one or more of requesting status information from the entities, modifying management parameters of the entities, or enabling self-test modes of the entities. The system also includes stored management specification information listing, in compliance with a universal specification language having a common syntax for representing the attributes and operations of any arbitrary manageable entity, the attributes which relate to the entities' functioning and control, and the management functions of the entities. The management specification information may further list the attributes and operations of entities which are subordinate to other entities. The management specification information includes polling information in predetermined fields of the common syntax. The polling information includes fields for specifying a default rate and a maximum polling rate at which the values of attributes should be requested from the entities. The management specification information may also include partition information in predetermined fields of the common syntax, the partition information representing groups of attributes having common data types. The management specification information may also include aggregation information in predetermined fields of the common syntax. The aggregation information represents groups of attributes having related functions in the management of the entity.

The management specification information may also include command information in predetermined fields of the common syntax, the command information lists the management functions which the entities are adapted to perform, the structure of the commands to be issued to the entities, and the structure of the replies to be received. The structure of the requests to be issued includes fields for listing arguments to a command. The structure of the replies to be received includes fields used for indicating the successful completion of the requested operation. The structure of the replies to be received includes fields used for indicating error conditions causing unsuccessful completion of the requested operation.

At least one management module includes an access module implementing protocols for communicating with one or more entities. The protocols are consistent with Ethernet standards or DECnet Phase IV standards, or DECnet Phase V standards.

Each command includes fields listing at least a related entity and operation, and the kernel includes a dispatcher for receiving and forwarding commands based at least in part on the entity and operation listed therein. The table of dispatch pointers comprises a directed graph of data structures, successive data structures in the graph corresponding to fields of the commands. The dispatcher includes a parser for parsing the directed graph in accordance with the entity and operation listed in a command to locate a terminal data structure having a dispatch pointer. The directed graph includes wildcard flags and successive data structures which may correspond to any value in a particular field of a command. The directed graph includes ellipsis flags and successive data structures which may correspond to any number of values in fields of commands. The parser includes a best-match unit for determining the most exact match for fields of a command, by searching first for exact matches for fields and then for wildcard matches for fields, or by searching first for exact matches for fields, then for ellipsis matches for fields.

The system includes a presentation device for displaying information to a user and receiving commands from a user, the commands and information being in specific predetermined formats. A presentation module receives commands from the presentation device and forwards information to the presentation device, the presentation module including conversion code to convert information received from an entity into a predetermined format for the presentation device, and forwarding code for forwarding commands from the presentation device to the dispatcher. The presentation module includes user interface information defining modes in which users interact with the system. The user interface information includes help information for providing information to the user on how to use the system. The user interface information includes graphic mode information defining pop-up menu contents and command line parse tables.

The kernel also includes a class database defining the different management information available from the respective entities. The presentation module includes menu generation routines for extracting data from the class database and generating menus of valid commands for display to the user. The menu generation routines are adapted to determine information relating to the configuration of said assemblage and generate menus of available entities for display to the user.

In general, in another aspect, the invention features a management module adapted to be stored for carrying out management functions by independently interpreting and executing selected management-related commands, for use in a system for controlling and carrying out management functions over an assemblage of entities. In preferred embodiments, the module includes dispatch pointers pointing to the module and associated with commands which are interpreted and executed by the module.

In general, in another aspect, the invention features a system for retrieving management information about an assemblage of entities in response to commands specifying a time schedule, wherein the entities interface within the assemblage for control of primary information handling functions and the entities further interface with the system to permit the accessing of the management information. The system includes storage containing records relating to the management information, each record including an indication of an associated time, and an information manager for retrieving management information contained in the records or accessing management information from the entities in response to a command, comprising a scheduler for possibly issuing a succession of subsidiary accesses or retrievals corresponding to the command at possibly multiple times according to the time schedule.

Preferred embodiments of this aspect include the following features.

A historical data recorder periodically accesses and stores new management information in the records in response to a predetermined schedule. The system is adapted to respond to a command specifying at least one desired time range, the time range possibly including past, present and future times, and the information manager includes means for satisfying the command by retrieving management information contained in the records, if possible, and otherwise accessing information relating to the specified time range from the entities. The information manager is configured to satisfy a command having a time range which includes all times prior to a specified time, by retrieving any record which is stored in the records during the time range, or otherwise accessing the information from the entities. The information manager is configured to satisfy a command by immediately accessing management information from the entities. Events occurring within the network are treated as a component of the state of the network and are stored in the records.

In general, in another aspect, the invention features a system for controlling and carrying out management functions over an assemblage of entities, wherein the entities interface within the assemblage for control of primary information handling functions and the entities further interface with the system to permit the carrying out of the management functions. The system includes stored management modules adapted to carry out the management functions by independently interpreting and executing selected management-related commands, storage containing domain specification information defining groups of entities, and a kernel adapted to issue commands to all entities of one the group by issuing individual commands to appropriate management modules.

Preferred embodiments of this aspect include the following features. The domain specification information complies with a universal specification language having a common syntax for representing any arbitrary group of entities. The common syntax provides for the incorporation of entities from a first domain into a second domain by reference to the first domain. The common syntax provides for the creation of subdomains of entities wholly contained within other domains. At least one management module comprises a domain management module for establishing and maintaining the domain specification information. The domain management module is responsive to commands for one or more of adding or deleting entities from groups, creating groups, or deleting groups. The domain management module is responsive to commands having filter procedures selecting entities of one or more particular domains. The filter procedures may select entities of subdomains wholly contained within other domains.

In general, in another aspect, the invention features a system for controlling and carrying out management functions over an assemblage of entities, wherein the entities interface within the assemblage for control of primary information handling functions and the entities further interface with the system to permit the carrying out of the management functions. the system includes stored management modules adapted to carry out the management functions by independently interpreting and executing selected management-related commands and issuing other commands to other modules, each command listing, in conformance with a common command syntax, the identity of the related entity and the operation to be performed, and a kernel comprising a table of dispatch pointers for directing the commands to the respective modules in which they are to be interpreted and executed.

Preferred embodiments of this aspect include the following features. The common command syntax provides fields for specifying subordinate entities, attributes, and operations. A first category of the management modules includes functional modules adapted to provide functional manipulation of data provided by the entities, and a second category of the management modules includes access modules adapted to implement the protocols for communication with the entities. The table of dispatch pointers includes a functional-access aspect facilitating communication between modules of the first category and other modules of the first category or modules of the second category. The system includes presentation modules adapted to receive commands from and forward information to the user using the primary information handling functions of the entities. The table of dispatch pointers includes a presentation-functional aspect facilitating communication between the presentation modules and modules of the first category. One module of the first category comprises a control functional module for communicating received commands directly to modules of the second category.

In general, in another aspect, the invention features a system for controlling and carrying out management functions over an assemblage of entities, wherein the entities interface within the assemblage for control of primary information handling functions and the entities further interface with the system to permit the carrying out of the management functions. The system includes stored management modules adapted to carry out the management functions by executing selected management-related commands, and at least one module storing rules identifying selected alarm conditions and comprising a rule generator for generating rules for storage and an alarm condition detector for detecting an alarm condition in response to the contents of the rules.

Preferred embodiments of this aspect include the following features. The management modules are adapted to carry out the management functions by independently interpreting and executing selected management-related commands. At least some management functions generate management information indicating the status of the primary information handling functions of one or more entities. The rules specify values for the management information at one or more times. The system includes storage containing records of the management information, each record including an indication of an associated time. The system includes an historical data recorder for periodically accessing and storing new management information in the records in response to a predetermined schedule.

In general, in another aspect, the invention features a system for controlling and carrying out entity management functions over an assemblage of entities and also controlling and carrying out self-management functions over itself, wherein the entities interface within the assemblage for control of primary information handling functions and the entities further interface with the system to permit the carrying out of management functions. The system includes at least one stored management module adapted to carry out the entity management functions by independently interpreting and executing selected commands, and further adapted to carry out the self-management functions on itself by interpreting and executing other commands, and a kernel comprising a table of dispatch pointers for directing the entity and self management commands to the respective modules in which they are to be interpreted and executed.

Preferred embodiments of this aspect include the following features. Each entity management command lists, in conformance with a common command syntax, the identity of the related entity and the operation to be performed, and each self-management command lists, in conformance with the common command syntax, the identity of the related module and the operation to be performed. The kernel includes a dispatcher for receiving and forwarding commands based at least in part on the operation and entity or module listed therein.

Brief Description of the Drawings

This invention is pointed out with particularity in the appended claims. The above and further advantages of this invention may be better understood by referring to the following description taken in conjunction with the accompanying drawings, in which:

Fig. 1A is a functional block diagram of a control arrangement constructed in accordance with the invention;
Fig. 1B is a block diagram of the information stored in the storage element of Fig. 1A;
Fig. 2A is a functional block diagram of a portion of the control arrangement depicted in Fig. 1A, particularly defining an entity comprising the control arrangement;
Fig. 2B illustrates the structure of a management module.
Figs. 3A through 3D define the management specifications defining the management view provided by functional modules and access modules comprising the control arrangement depicted in Fig. 1A, and Fig. 3E defines the dispatch specifications for the functional modules and access modules;
Fig. 4 depicts the structure of a data dictionary which includes information defined by the management specifications shown in Figs. 3A through 3D;
Figs. 5 and 6 are functional block diagrams depicting various modules and data structures in the control arrangement depicted in Fig. 1A;
Fig. 7A depicts the parameters used in requests generated by the presentation modules and functional modules in the control arrangement depicted in Fig. 1A;
Fig. 7B depicts the structure of time context handles and context blocks used by the request of Fig. 7A;
Figs. 8A and 8B depict data structures in dispatch tables used by a dispatcher as depicted in Figs. 5 and 6 in connection with processing of requests from the presentation modules and functional modules in the control arrangement depicted in Fig. 1A;
Figs. 9A and 9B depict the operations of a dispatcher in connection with its associated dispatch table in processing a request from a presentation module or a functional module;
Fig. 9C depicts the format of a configuration and domains database;
Fig. 10A depicts the structure of a functional module used in establishing and detecting alarm conditions, and Fig. 10B depicts the structure of rules used in establishing alarm conditions.

GENERAL DESCRIPTION

Fig. 1A depicts a functional block diagram of an arrangement constructed in accordance with the invention for controlling and monitoring the status and condition of a complex system. (The complex system itself is not shown.) Preliminarily, one example of a complex system controlled by the arrangement depicted in Fig. 1A includes a distributed digital data processing system, comprising a plurality of nodes, including individual computers, terminals, terminal servers and other components, which communicate by means of messages transmitted over a network. One example of such a digital data processing system is described in U. S. Patent Application Serial 06/616553 filed on June 1, 1984. It will be appreciated, however, that the control arrangement depicted in Fig. 1A is not limited to the control of a distributed digital data processing system, but may be used to control a number of diverse types of complex systems.

Such complex systems are challenging to manage, particularly because the status and capabilities of the complex system are constantly changing. Therefore, the management arrangement and the management functions it provides must also change to adapt to new management requirements of the system. As will be discussed in detail later, the arrangement of Fig. 1A features extensibility, which allows the arrangement to adapt efficiently to changes in the complex system.

For the purposes of this document, the components of the complex system will be called entities. Entities are discussed in terms of classes and instances. An entity class defines entities of a particular type, e.g. one class would include all local area network bridges from a given vendor. Each entity is a member of a class, and forms an instance of that class.

With reference to Fig. 1A, the control arrangement includes several types of control modules, including presentation modules 10A through 10K (generally identified by reference numeral 10), functional modules 11A through 11M (generally identified by reference numeral 11) and access modules 12A through 12N (generally identified by reference numeral 12). The presentation modules 10 generally provide the user interface for the operators providing control for the complex system, including control of terminals used by the system operators. Each functional module 11 generally provides management control and monitoring in connection with a class of functions. Each access module 12 generally provides management control for a particular type of controllable entity, in a set belonging to a class of controllable entities, in the complex system. The presentation modules 10 communicate with functional modules 11 through a presentation-function aspect of a kernel 13, 14, hereafter called simply the presentation-function kernel 13, and the functional modules 11 communicate with the access modules through a function-access aspect of the kernel 13, 14, hereafter called simply the function-access kernel 14.

The functions that are required from control modules 10, 11, 12 may vary widely depending upon the topology of the complex system being managed. Therefore, to provide the arrangement with adaptability and extensibility, control modules 10, 11, 12 may be dynamically added or removed from the arrangement to adapt the arrangement to the topology of a particular complex system, and to the changes in that topology.

To further the goals of adaptability and extensibility, the control modules 10, 11, 12 form a "division of labor" for the tasks to be performed in management of the complex system. In this way, the tasks associated with, e.g., the management protocols of a distributed data processing system, may be separated from the tasks associated with, e.g., the display of management information to the user.

A. PRESENTATION MODULES

More specifically, the presentation modules 10 provide presentation services, which may comprise, for example, support for a user interface such as a video display terminal, personal computer or computer workstation, which may be used by a system operator to control the operation of the various functional modules 11 and access modules 12, thus controlling and monitoring various entities in the complex system. The presentation services are required independently of the management functions or the entities which are managed by the system depicted in Fig. 1A, and thus are provided regardless of the nature of the management functions or entities. Each operator interface or terminal may be controlled by a plurality of presentation modules 10. The various presentation modules 10 control diverse aspects of the operator interface, including such details as, for example, icons, menus, graphics and support for displaying and parsing a command line. Other presentation modules 10 provide specific output support for various types of graphical displays, for example, histograms, bar charts, pie charts, or other types of pictorial representations to be displayed on a terminal screen for an operator. Still other presentation modules 10 provide transfer of management requests, which may be noted by means of icons, menus, graphics or commands which the operator entered on the command line, to the presentation-function kernel 13, and of management information from the presentation-function kernel 13 for display on the video display terminal used by an operator.

B. FUNCTIONAL MODULES

The functional modules 11 are associated with, and generally support, the specific management applications provided by the control arrangement depicted in Fig. 1A. The management applications exist independently of the presentation services provided by the presentation modules 10 (other than to the extent that the presentation modules 10 notify an operator of the management applications that are provided by the control arrangement) and the particular entities comprising the complex system that are being managed by the control arrangement.

A management application which could be provided by a functional module 11 would, for example, analyze the communications load in a distributed data routing system. To perform such an analysis, a functional module would access communications data, such as the number of packets sent and the number of bytes sent, from several of the entities of the distributed routing system. The functional module would then collate the information into higher-level information, such as the average packet size and the percent utilization of the communications resources of the routing system. This information would then be forwarded to the user or made available to other functional modules in the execution of other management applications.

As seen in the above example, a functional module "adds value", in the form of data collation or correlation services, to management information that is available from the complex system. In addition, functional modules may make use of data produced by other functional modules to perform high-level services for management of the complex system.

In one specific control arrangement for controlling a distributed digital data processing system, one functional module 11, for example, manages the topology of the network and shows the topology to an operator through a presentation module 10.

Another functional module 11 may comprise a configuration functional module that, for example, defines the configuration, that is, the various entity instances and their inter-relationships, of the distributed digital data processing system, permits an operator to control the configuration of the network, by enabling nodes and other entity instances to be added to or deleted from the network, changes access rights by the various users of the nodes, and also maintains a configuration (or instance) database by which the operator can determine the changes to the configuration of the network over time.

Another functional module 11 in the control arrangement may, for example, control various alarms indicating occurrence of selected events in the distributed digital data processing system; this alarm functional module 11 monitors the status and condition of various entities in the distributed digital data processing system and generates an alarm indication to an operator, through the appropriate presentation module 10, in response to the status or conditions having selected values to advise the operator thereof.

Yet another functional module 11 may, for example, establish domains of entities in the distributed digital data processing system, to limit the purview of control or monitoring by an operator or to simplify control or monitoring by the operator.

Another functional module 11 may, for example, function as a historical data recorder functional module 11 to periodically poll various entities in the complex system to determine their values at specific times and establish and maintain a database of the times and values to facilitate generation of usage statistics.

Yet another functional module 11 may not control any specific aspect of the complex system, instead operating as a pass-through to permit an operator to control or monitor primitive functions of the complex systems directly through the access modules 12.

A management application may require the services and operation, in particular sequences, of a number of access modules 12, and the functional module 11 which supports the management application coordinates the sequencing of the operations by the various access modules 12 that are required to accomplish the management application. In addition, a management application provided by one functional module 11 may require the application of additional functional modules 11 in the control arrangement, which the one functional module may also coordinate.

The functional modules 11 are invoked, initially, by the presentation-functional kernel 13 in response to management requests entered by an operator obtained by a presentation module 10. A functional module 11 may also be invoked by a request directly received from another functional module 11. In addition, a functional module 11 may generate a request for processing by an access module 12.

C. ACCESS MODULES

The access modules 12 are associated with, and support, the various primitive management operations provided by the control arrangement in connection with the various entities comprising the complex system managed by the control arrangement depicted in Fig. 1A. For example, in a distributed digital data processing system, the entities may comprise, not only the various hardware components of the system, including various computers, disk and tape storage units, routers, and so forth, which may comprise nodes in the distributed digital data processing system, but also software components including virtual circuits, databases, and so forth. The access modules 12 are invoked by the functional-access kernel 14 in response to requests from a functional module 11.

Access modules 12 for controlling and monitoring a distributed digital data processing system may control several different types of nodes or different levels in the message transfer protocols used by the nodes to generate and transfer messages. One access module 12 may, for example, control and monitor the status of various portions of a bridge that links two local-area networks together, permitting messages to be passed between nodes on the two local area networks. Such an access module 12 may, for example, initialize the bridge and enable it to start operating, disable the bridge, monitor its end-to-end operation, determine the number of message passing buffers it has and determine whether it has sufficient buffers to operate effectively in the system.

Another access module 12 may control and monitor the operation of the message generation and decoding portions of the various nodes of the distributed digital data processing system, the virtual circuits, sessions and other links established between nodes, various timers and counters indicating activity or inactivity thereover and so forth. Similarly, another access module 12 may control and monitor the operation of the nodes' network layer portions, which control the actual transmission and reception of messages over the network, including various message transmission and reception counters, transmission and reception timers, and so forth. Access modules 12 controlling both of these may also be used, in addition to monitoring the values of the various timers and counters, to establish limits on the number of concurrent virtual circuits and sessions that a node may maintain and establish other default and operational parameters.

In specific embodiments, access modules may provide for access to management functions at ETHERNET LAN bridges, connectivity test or IEEE 802 functions ETHERNET stations, port segmenting control and check functions at ETHERNET repeaters, or management functions at FDDI entities. In addition, access modules may provide for access to management support at DECnet Phase IV or Phase V nodes, or DEC Terminal Servers, as promulgated by Digital Equipment Co., Maynard, MA.

D. REQUESTS

The control modules 10, 11, 12 interact with each other and with the user through requests. Requests are of two general types. A request may, for example, enable something to occur in the complex system, that is, it may cause the state or condition of the complex system to be changed. In processing such a request, one or more access modules 12 perform predetermined operations that change the state or condition of one or more entities in the complex system being managed. The access modules 12 that process such a request generate status information indicating the status of the request, which they return to the functional-access kernel 14.

Alternatively, a request may solicit information as to the status or condition of one or more entities in the system, the entities being identified in the request. In processing such a request, one or more access modules 12 may determine the status or condition of the entities, and return an identification thereof to the functional-access kernel 14. In other cases, information stored in the control arrangement (such as by a historical data recorder functional module) may be used to satisfy the request.

In addition, a request may be of both types, that is, it may change the state or condition of one or more entities, and may also request information as to the states or conditions of the entities after the change. In processing such a request, the access modules 12 cause the change to occur, if possible, and return status information as to the status of the request, as well as information as to the states or conditions of the entities.

Requests may be generated in response to an operator action at a terminal presentation device. In that case, the presentation module 10 controlling the terminal generates a request, which it transmits to the presentation-functional kernel 13. In addition, requests may be generated directly by appropriate functional modules 11. For example, a functional module 11 operating as a historical data recorder may generate requests to periodically determine the status or conditions of the respective entities in the complex system for storage in a historical database for use in later processing if required by an operator.

E. KERNEL

The kernel 13, 14 includes several elements, including an information manager 15, 20 (hereafter referred to simply as information manager 15 or information manager 20, which form one and the same information manager), a dispatcher 16, 21 (hereafter referred to simply as dispatcher 16 or dispatcher 21, which form one and the same dispatcher) and a data storage element 17, 22 (hereafter referred to simply as data storage element 17 or data storage element 22, which form one and the same data storage element, as described below.

F. DATA STORAGE

The data storage element 17, 22 may comprise one or more high speed RAM's containing dispatch data structures, or one or more fixed disk drives or other storage means, according to the types and amount of data stored therein. In addition, data of different types may be stored in various storage means for later use by the kernel, all of these means being represented diagrammatically by the single data storage element 17, 22.

Referring to Fig. 1B, in one embodiment, the data storage element 17, 22 maintains information as to the existence and condition of the various entities comprising a complex system at various points in time, in particular, selected information as to the status and conditions of various entities controlled by the access modules 10 as obtained by the historical data recorder functional module 11. This is stored in a historical database 26.

Other information may also be stored in data storage element 17, 22. In particular, as discussed above, a configuration module may form a configuration database 23 indicating the presence of entity instances in the complex system. A domains module may store a database 25 describing domains of entities for use in limiting the user's scope of control. Alternatively, the domain information may be stored as an element of the configuration database 23. Also, an alarms module may use an alarm rule base 24 to verify alarm conditions within the complex system.

Other information, which relates to the individual modules in the control arrangement may also be maintained in storage element 17, 22. For example, as will be detailed below, a dispatch table 28 for use by the dispatcher 16, 21 may store the locations of the modules and the operations, entities, and attributes which they service. In addition, the control arrangement may maintain a data dictionary 27 storing the attributes, directives and sub-entities of each of the various classes of entities in the complex system. This latter information may be used to, e.g., process requests from the user and/or to create menus to prompt user requests.

G. INFORMATION MANAGER

Referring to Fig. 1A, as described in detail later, if the information manager 15 receives a request from a presentation module 10 to which it can respond using the information in the data storage element 17, it intercepts the request and generates a response to the request, which it transmits to an appropriate presentation module 10 for display to the operator which provided the request. If the information manager 15 is unable to respond to the request, it then determines whether the request relates to the current time or a time in the future; that is, the information manager 15 determines whether the request should be processed immediately or scheduled for a specified time in the future. At the appropriate time, whether immediately or at the scheduled time, the information manager 15 transfers the request to the dispatcher 16. From the nature of the request, the dispatcher 16 identifies a functional module 11 to process the request, and transfers the request to that functional module 11.

In response to the receipt of a request from the dispatcher 16, the functional module 11 proceeds to process the request. It may, in response to the request, initiate one or more operations, each represented by a request, hereafter called a subordinate request, which it directs to another functional module 11 or to the functional-access kernel 14. Upon receiving responses to all of the subordinate requests, the functional module 11 generates a response which it transmits to the dispatcher 16. The dispatcher 16 then formulates a response that it transmits, through the information manager 15, to the appropriate presentation module 10 for display to an operator.

The functional-access aspect of kernel 14 includes the information manager 20, the dispatcher 21 and the data storage element 21. A subordinate request from a function module 11, directed to the function-access kernel 14, is received initially by the information manager 20. The data storage element 22 also contains information, as provided by the historical data recorder functional module 11, as to the condition of the complex system at various points in time, in particular, selected information as to the status and conditions of the various entities controlled by the access modules 10.

If the information manager 20 receives a subordinate request from a function module 11 to which it can respond using the information in the data storage element 22, it intercepts the request and generates a response to the subordinate request, which it transmits to the function module 11 from which it received the subordinate request. If the information manager 20 is unable to respond to a subordinate request from a functional module 11, it then determines whether the request relates to the current time or a time in the future; that is, the information manager 20 determines whether the request should be processed immediately or scheduled for a specified time in the future. At the appropriate time, whether immediately or at the scheduled time, the information manager 20 transfers the subordinate request to the dispatcher 21. In response to the receipt of a subordinate request from the information manager 20, the dispatcher 21 identifies an access module 12 to process the subordinate request and transfers the subordinate request to that access module 12.

In response to the receipt of a subordinate request from the dispatcher 21, the access module 12 proceeds to process the request. It may, in response to the subordinate request, initiate one or more operations in connection with the entity of the complex system controlled thereby. If the subordinate request requires the access module 12 to change the state or condition of the entity, it attempts to do so and generates a response containing status information indicating the status of the attempt, that is, for example, whether the change was successful, unsuccessful, or partially successful. On the other hand, if the subordinate request requires the access module 12 to identify the state or condition of the entity, it generates a response indicating the entity's state or condition. Finally, if the subordinate request requires the access module 12 to do both, it attempts to change the state or condition of the entity and generates a response indicating the status of the attempt and also the entity's new state or condition. In any case, the access module 12 transmits the response to the dispatcher 21, which transfers it to the functional module 11 which generated the request. The functional module 11 uses the response from the access module 12 in formulating its response to a request from the dispatcher 16 or to a subordinate request from another functional module 11, as appropriate.

A functional module 11, upon receiving a subordinate request from other functional modules 11, processes it in the same manner as it processes a request from the dispatcher 21.

H. ADVANTAGES

The control arrangement depicted in Fig. 1A provides a number of advantages. The control arrangement essentially forms a processing chain, with each element along the chain attempting to process a request before passing it along to the next element. Thus, if the information manager 15, 20 can process the request, based on the contents of associated data storage element 17, 22, without requiring further processing by another element further down the chain, It does so.

Furthermore, the control arrangement is extensible, so that additional presentation modules 10, functional modules 11 and access modules 12 can be easily added, as described below, without changing the architecture of the control arrangement. Addition of functional modules 11 and access modules 12 is by way of an enrollment procedure, which is described below in connection with Fig. 5. Additions or deletions of modules 10, 11 or 12 can be made merely by modifying, as described below, the contents of certain data structures in the data storage element 17, 22, and other data structures maintained by the presentation modules 10, as depicted in Fig. 5.

Additionally, the modular, extensible nature of the control arrangement facilitates management of the control arrangement itself. The same dispatch and request paradigms which are used to issue management directives to the complex system may also be used to issue commands to the management modules themselves. This eliminates the need for an additional management application to manage the control arrangement itself.

Also, as the functions of the modules are specified in a standard format and available to the control arrangement as a whole, the control arrangement can provide full user interface support for the modules, thus freeing module designers from the burden of supporting a user interface to each module. This type of "automatic" user interface support also guarantees a uniform look and feel to the user interface regardless of the source or nature of the management modules being used.

It will be appreciated that, if the control arrangement is used to control a distributed digital data processing system, it, including its various elements, may comprise a plurality of routines processed by the various nodes and computers comprising the distributed digital data processing system; that is, computer facilities, in addition to those comprising the distributed digital data processing system being controlled, are not required to process the modules comprising the control arrangement to control the distributed digital data processing system. Conventional procedure call mechanisms, interprocess communication mechanisms and inter-nodal communications mechanisms may be used to transfer communications, including requests, subsidiary requests and responses, between the various portions of the control arrangement which may reside in different parts of the same process, in different processes in the same node, and in different nodes. If the modules reside in different processes in the same node or in different nodes, interprocess and internode communications mechanisms as depicted in Fig. 6, described below, are used to transfer requests and subsidiary requests, as well as responses, among the various processes and nodes.

I. ENTITY MODEL

Before proceeding further, it will be helpful to describe further the relationship between the control arrangement depicted in Fig. 1A, and the complex system being controlled. Specifically, referring to Fig. 2A, the control arrangement comprises a director 35, which includes all of the presentation modules 10, the functional modules 11, and the access modules 12, along with the kernel 13, 14. The complex system includes one or more entities 36. Each entity 36 includes a service element 31, a management interface 30 and a service interface 33. The management interface controls and monitors the service element through an agent 34. The service element is the actual managed portion of the entity 36 and provides the entity's primary function or function. That is, the service element 31 performs the function of the entity required within the context of the distributed digital data processing system. If, for example, the entity performs communications over a network for a node, the service element 31 performs the communications.

As noted above, the service element 31 is managed through an agent, which communicates with the director, specifically, with the access modules 12, through the management interface 30 and the service interface 33. The communications through the management interface 30 facilitates turning the service element 31 on or off and its initialization, and also permits the director 35 to determine the operational status of the entity 36. Communications through the service interface 33 permits the director 35 to control and monitor service element 31 otherwise, by, for example, establishing conditions of selected attributes, such as communications parameters in the case of an entity 36 which performs communications, in context of controlling the entity 36, or determining the values of counters, in the context of monitoring the entity 36.

The management of an entity is characterized by the directives it supports, and its attributes, which are, broadly, those parameters which relate to its functioning and control and are associated with directives. For example, if the entity is a router which communicates data packets through a distributed data processing network, the attributes of the router may include the number of packets transmitted, and the number of bytes transmitted. If the entity is a modem, the attributes may include the counters and status registers which relate to the modem operation. Examples of directives include SHOW, which will retrieve attribute values, and SET, which modifies attribute values.

The service interface relates to the function of the entity, and the management interface relates to operation of the agent. The directives and attributes which are accessed through the service interface characterize the function of the entity, whereas the directives and attributes which are accessed through the management interface characterize the control and monitoring of the entity.

To clarify the roles of the two interfaces, and to provide an example of how the above model applies to a particular entity, consider a controllable entity which is a modem. The modem may have several functional attributes, such as the baud rate, line selection, and power switch setting. In addition, the modem may have several management attributes, such as its the percent utilization of its lines and the time elapsed since the last self-test. The baud rate, line selection, and power switch setting relate to the immediate operation of the modem, and as such would be accessed through the service interface. The percent line utilization and time elapsed since the last self-test to the general operation of the modem, and as such would be accessed through the management interface.

To elaborate on the above example, note that the presentation modules, during presentation of management information on a presentation device, use the service interface of the presentation device, because the presentation of information is the main service of the presentation device. However, an access module in the control arrangement may also manage the presentation device, for example by polling it to determine if it is turned on.

In addition to the attributes discussed above, there are other "pseudo-attributes" which relate to the entity but are not stored by the entity as such. Pseudo-attributes generally are attributes which are required by the entity model description but not supplied by the entity. An example is the attribute IMPLEMENTATION, which may be the synthesis of the attributes IMPLEMENTATION TYPE and VERSION supplied by the entity, and the CREATION TIME of the entity. Pseudo-attributes are maintained by the access module which is responsible for accessing the entity.

It is worth noting at this point that the entity model is a generalized method for describing directives and attributes of an entity, and does not imply any structure within the entity itself. The entity model is a tool which allows the control arrangement to refer to the operations and attributes of any arbitrary entity in a consistent fashion. Any arbitrary entity may be "plugged into" and managed by the control arrangement of Fig. 1A by (1) describing it consistent with the entity model, (2) implementing an appropriate access module, and (3) plugging (enrolling) the access module into the control arrangement.

J. MANAGEMENT OF MANAGEMENT MODULES

As noted above, in a control arrangement which controls a distributed digital data processing system, the various presentation modules 10, functional modules 11, access modules 12 and kernel 13, 14 are processed by the various nodes comprising the distributed digital data processing system. In that case, the various modules 10, 11 and 12 and kernel 13, 14 form entities in the complex system, and may be controlled in the same manner as other entities, as described above.

The dispatch and request paradigms which are used to issue management directives to the complex system are also used to issue commands to the management modules themselves. As will be seen in the dispatch specifications below, in addition to management routines for managing the complex system, each module contains self-management routines which manipulate the internal attributes of the module. Both the external and internal routines may be accessed by requests using the request syntax. Therefore, as the capabilities for management of the complex system are increased by addition of new control modules, the capabilities for management of the control arrangement are similarly increased.

SPECIFIC DESCRIPTION

A. MANAGEMENT MODULE STRUCTURE

1. Overview

Referring to Fig. 2B, in one particular embodiment, the structure of a management module includes executable code 38 that implements the management functions provided by the module. In particular, for an access module, the executable code includes the access protocols for the entity classes which are serviced by the access module. For a functional module, the executable code includes instructions for computing the higher-level functions provided by the module. For a presentation module, the executable code includes the interface protocols for the presentation devices supported by the presentation module.

The module may require private memory to store various read-only or read/write variables relating to the module's function. This storage is provided to the module as an allocated region 32. This storage may be used, for example, by a presentation module to store parse tables or presentation forms data, or by an access module to store password information in a wildcarded request (see below).

The access points of the various procedures provided by the access modules are indicated by pointers in the dispatch entries 39A and 39B. As will be more fully discussed later, the dispatch entries are merged into the dispatch table stored in the kernel storage 17, 22, and are used to locate the various procedures which the module supports. As shown in Fig. 2B, dispatch pointers 39A relate to the procedures in the module which provide management services to the complex system, whereas dispatch pointers 39B relate to the procedures in the module which provide management services to the module itself. As discussed above, when the module is enrolled into the control arrangement both sets of pointers are loaded into the kernel memory for use in managing the complex system or the modules which comprise the control arrangement.

In addition to the above structure, the module is associated with a management specification 48 which describes the classes of entities and attributes which are serviced by the module, as well as the directive and response structure for requesting services from the module. The management specification also specifies the management of the module itself. During the enrollment of a module, the related management specification is loaded into the data dictionary.

2. Management Specification

The properties, composition and structure of the service element 31 and service interface 33 of the entities of the complex system being managed by the control arrangement (Fig. 1A), as well as the various entities comprising the control arrangement, are defined by a management specification and dispatch specification. Figs. 3A through 3D detail the management specification for an entity, and Fig. 3E defines a dispatch specification which is used in initiating a particular operation in connection with the entity. With reference initially to Fig. 3A, the management specification for an entity includes a header portion 40 and a body portion 45. The header portion 40 includes certain identification information such as a name field 41 which contains a name that identifies the entity, a version field 42 which contains a version identification, a facility field 43 containing location information indicating the location of the entity within the complex system (for example, the identification of the node if the complex system is a distributed digital data processing system), and a type declaration field 44 which indicates selected data type information.

In an alternative embodiment, the header portion may also include a symbol-prefix field which is used in conjunction with the symbol field 52, discussed below.

The body portion 45 of the management specification contains the actual management specification for the entity.

The body portion 45 is further defined in Fig. 3A. Preliminarily, the control arrangement includes two general types of entities, namely, a global entity, and a subordinate entity. The control arrangement facilitates a hierarchy of entities, as defined above, with the global entity identifying a top level entity in a hierarchy and a subordinate entity identifying a entity that is subordinate to another entity in the hierarchy. The body portion 45 of a management specification includes one of two types of entity definitions, that is, a definition 45A to a global entity or a definition 45C to a subordinate entity.

A management module may provide services to a global class of entities, or to a class of subentities within a global entity class. A particular example occurs in the DECnet Phase IV, as promulgated by Digital Equipment Corporation, Maynard, Massachusetts; in DECnet Phase IV, Adjacent_Node is a subordinate entity class, whose superior entity class is Node4_Circuit. If a management module provides services specifically to the Adjacent_Node subordinate entity class, the management specification must provide a mechanism to indicate that the management specification for the global class resides in the management specification for another module (that which manages the Node4_Circuit class).

The definitions 45A and 45C to a global and subordinate entity, respectively, are further defined in Figs. 3A through 3D. An entity definition 46 includes a name field 47 that includes a name and a code by which the entity can be identified. In addition, the name field 47 identifies the entity as a global or subordinate entity and identifies a class name for the entity. If the entity definition is for a subordinate entity, it has a superior field 50 which identifies the superior entities in the hierarchy. An identifier field 51 includes a list of attribute names for attributes which are defined later in an entity body portion 53. Finally, a symbol field 52 includes a symbol that is used to generate a specific compiler constants file which contains consistent names for use by an entity developer.

In an alternative embodiment, a DYNAMIC field may be included in the entity definition. This field may have the values TRUE or FALSE, and indicates whether the management specification for the entity should be stored in the configuration database (Fig. 1B). This provides the management module developer a way to indicate precisely which subordinate entity instances are to be stored in the configuration database. In this way, entities such as connections between nodes which are highly dynamic do not need to be stored in the configuration of the system. This aliminates the overhead caused by repeatedly adding and deleting dynamic instances. The boolean value of the DYNAMIC field indicates if the entity class is dynamic in nature; if TRUE, instances of the entity class will not be stored in the configuration, if FALSE, instances of the entity class will be stored in the configuration.

As noted above, an entity definition 46 for an entity includes a body portion 53. The body portion 53 is defined in detail in Fig. 3B. With reference to Fig. 3B, the body portion 53 of a management specification includes four portions, namely, an attribute partition definition list 54, an aggregation definition list 55, a directive definition list 56 and a subordinate entity list 57, if the entity class contains any subordinate entities. If the body portion 53 includes a subordinate entity list 57, each item in the subordinate entity list 57 comprises an entity definition 46 (Fig. 3A), with the name field 47 including "SUBORDINATE".

As mentioned above, the entity body contains an attribute partition list 54 and an attribute aggregation list 55. It is useful at this point to explain the distinction between these lists. Each list takes the entity's full set of attributes and associates each attribute with one or more groups; the groupings set forth by the partition list 54 are independent from those set forth by the aggregation list--each list is an independent characterization of the entity's attributes.

The partition list 54 identifies and groups all attributes having similar form; for example, an attribute partition may include all counters or all status attributes (flags). The word "partition" is used to indicate that the groups formed by attribute partitions are true partitions of the attributes--no attribute may be a member of two partitions, and each attribute must be a member of exactly one partition.

The aggregation list 55 identifies and groups all attributes having similar function. For example, an access module for a NODE4 global entity class may define an attribute aggregation called "SQUERGE". The SQUERGE attribute aggregation may include all attributes relating to the current operational performance of a NODE4 class entity, e.g., a counter type attribute indicating the number of bytes sent, and characteristic type attribute indicating the pipeline quota. In this example, a user could then view these statistics together by a command such as:

```
SHOW NODE (instance) ALL SQUERGE
```

The word "aggregation" is used to indicate that aggregations contain attributes with like function, but do not necessarily form partitions of the attributes. One attribute may be a member of more than one aggregation, and all attributes do not need to be a member of an aggregation.

The attribute partition definition list 54 includes one or more attribute definitions 64 as further defined on Fig. 3B. Each attribute partition definition 64 includes a kind field 56 which identifies the attribute as being of a particular type, including an identifier type attribute, a status type attribute, a counter type attribute, a characteristic type attribute, a reference type attribute or a statistic type attribute. For each type of attribute, the data type is provided by an appended field 68. The attribute partition definition 54 may also include fields 60 and 61 which indicate, respectively, a default polling rate and a maximum polling rate for the entity. As noted above, a historical data recorder functional module 11 may periodically obtain status and condition information for storage in the data storage element 17, 22 in connection with the various entities comprising the complex system. The contents of the polling rate fields identify the default and maximum rates at which the respective entities will provide status and condition information. In addition, an attribute definition includes one or more attribute fields 62 each including an attribute name 63, which includes a code by which the attribute may be accessed, and an associated attribute body 64.

All definitions for attributes which are members of a partition reside within one partition definition 54 as set forth above. The independent aspects of the attributes are set forth by one of more attribute body definitions 64. Fig. 3B further describes the information contained in an attribute body 64 in an attribute field in an attribute partition definition 55. An attribute body 64 may include a number of fields, including an access information field 65 which indicates whether the attribute can be read or written and a display field 66 which indicates whether the attribute should be displayed to an operator, by means of a presentation module 10. A default value field 67 identifies a default or initial value for the attribute. A symbol field 70 contains a symbol that is used to generate a specific compiler constants file which contains consistent names for use by an entity developer.

An attribute body 64 further includes a categories field 71 which identifies one or more categories with which the attribute may be associated. If the complex system is a distributed digital data processing system, the categories may include but need not be restricted to categories defined by the 74-98-4 Open Systems Interconnect (OSI) standard, including CONFIGURATION, FAULT, PERFORMANCE, SECURITY or ACCOUNTING. In addition, the attribute body 64 may include polling rate information in fields 72 and 73 if the polling rates for the particular attribute defined by the attribute body 64 are different than the polling rates defined in fields 60 and 61 in the attribute partition definition 54. Finally, the attribute body 64 may include a private variable field 74 which identifies private variables that are used in the management module in processing relating to the attribute.

In an alternative embodiment, the polling rate information may be omitted entirely from the attribute definitions, owing to the implementation-specific nature of this data. In addition, in alternative embodiments, a UNITS field may be included in the attribute body 64. Where a UNITS field is included, numeric data types can (and should) have their units defined.

Attributes can be aggregated to simplify management of the complex system. The aggregation definition portion 55 of the entity body 53 identifies one or more aggregations which the entity includes. The contents of an aggregation definition portion 55 are defined in detail on Fig. 3B. An aggregation definition portion 55 includes an aggregation name field 75 which identifies the aggregation and an attribute list 81 identifying the attributes included in the aggregation. An aggregation definition portion 55 may also include a list of directives, that is, requests which may be processed by reference to the aggregation. An aggregation definition portion 55 may include a symbol field 77 similar to the symbol field described above, a categories field 80 that may contain but is not limited to OSI category information, and a private variables field 82 that identifies private variables used in processing relating to the attributes included in the aggregation identified by the aggregation name in field 75.

An entity processes directives which are generated by the control arrangement in response to the requests and subordinate requests from a presentation module 10 and a functional module 11, respectively. Each directive includes a directive request, which defines an operation to be performed, and may include a response and an exception which define responses that the entity may make in connection with the operation. Each directive is defined by a directive definition 56. Figs 3C and 3D detail the structures of a directive definition 56. With reference to Fig. 3C, a directive definition 56 includes a name field 83, which includes a code by which the directive can be identified and accessed. A directive includes a request definition field 90, which identifies the structure of a request or subordinate request, a response definition field 91 which defines the structure of a response, and an exception definition field 92 which defines the structure of an exception which may be generated during processing of the directive. The details of the fields 90, 91 and 92 will be described below.

A directive definition 56 may also include a field 84 which indicates whether the directive is an action directive, that is, whether it enables a change in the condition or status of one or more entities in the complex system, or whether it merely initiates return of status or condition information. In an alternative embodiment, the action field 84 may be replaced by a DIRECTIVE_TYPE field which indicates whether the directive is of the EXAMINE, MODIFY, or ACTION type. An EXAMINE directive operates on attributes only and does not modify; examples include SHOW or DIRECTORY directives. A MODIFY directive operates on attributes only and does modify; examples include SET, ADD, or REMOVE directives. An ACTION directive does not operate on attributes, rather, ACTION directives operate on the entity itself; examples include CREATE and TEST directives.

A field 85 may be provided to indicate whether the directive is accessible by a presentation module 10. An identifying text string may be provided in a symbol field 86. In addition, a categories field 87 may define, but need not be limited to, one or more OSI categories, as defined above in connection with field 71 (Fig. 3B).

The structure of the request definition field 90 in a directive definition 56 is defined in Fig. 3C. In addition to the word "REQUEST", the request definition field 90 may include zero or more arguments 91, each identified by a name field 92 including an access code. In addition, an argument may include a display field 93 that indicates whether the argument is to be displayed by a presentation module 10 to an operator. The argument nay also include field 94 which indicates whether an operator must provide a value for the argument, a default field 96 including a default value, a symbol field 97 including an identifying text string, and a units field 95 which identifies the units of measurement of the argument values. In addition, the argument 91 may include a private variable field 100 identifying the private variables used in processing in connection with the argument.

The structures of a response definition field 91 and an exception definition field 92 are depicted in Fig. 3D. With reference to Fig. 3D, a response definition field 91 includes a response name field 101, which also includes a code by which the response can be accessed. A severity field identifies whether the response indicates SUCCESS in performing the request defined by the request field 90, or whether the response is INFORMATIONAL. A text field 103 indicates a text string which the presentation module 10 can display to an operator to indicate the response. In addition, a response definition field can include one or more argument fields 104, each including a name field 105, a units field 106 and a symbol field 107.

In alternative embodiments, the SEVERITY field 102 may be replaced with a SYMBOL field containing an identifying text string for the response, and the ARGUMENTS field 104 may include a boolean DISPLAY field for indicating whether the response should be displayed to the user.

The structure of the exception definition field 92 is similar to that of the response definition field 91, including fields 111 through 117, which are similar to fields 101 through 107 of the response definition field 21. The severity field 112, however, can contain three values, including WARNING, ERROR and FATAL, indicating the severity of the error giving rise to the exception.

As in the response definition 91, in alternative embodiments, the SEVERITY field 112 may be replaced with a SYMBOL field containing an identifying text string for the response, and the ARGUMENTS field 114 may include a boolean DISPLAY field for indicating whether the response should be displayed to the user.

3. Dispatch Specification

Fig. 3E defines a dispatch specification 39A (Fig. 2B) which is used in defining initiation of particular operations by an entity. The information in the dispatch specifications for an entity are used to generate pointers to procedures to perform the operations. With reference to Fig. 3E, the dispatch specification includes a header 200 which defines the beginning of the dispatch specification and contains a table name, and a footer 201 that defines the end of the dispatch specification. Between the header 200 and footer 201, the dispatch specification includes one or more dispatch entries 202 each of which defines an operation in connection with one or more entities and attributes.

The dispatch entry includes a verb portion 203 and an entity entry 204, which together identify an operation. Effectively, the verb portion 203 and the entity portion 204 of the dispatch entry corresponds to a directive defined by the management. Directives may either operate on entities, or on attributes defined by an attribute portion 205 of the entity defined by an entity entry 204. The contents of the entity entry 204 correspond to an entity or sub-entity identified by an entity class and instance name in the name fields 47 and 50 of the entity definition 46. Similarly, the contents of the attribute portion 205 correspond to attributes that are defined by the name field 62 of the attribute definitions 54 of the entity body 53 of the entity definition 46.

The dispatch entry 202 also includes a procedure pointer portion 206, which contains a pointer to an entry point to a procedure in an access module which processes a directive in connection with the entity and attributes identified in portions 203, 204 and 205 of the dispatch entry 202. As will be described below in connection with Figs. 5, 7A and 8B, the dispatch specification is used in formulating data structures, specifically dispatch entries 134 (Fig. 8B) of dispatch tables 28 (Fig. 5) that are used by the kernel 13, 14 to transfer requests to the proper functional module 11 or access module 12 for processing. A request or subsidiary request essentially defines a verb, an entity and an attribute partition, and the kernel compares the verb, entity and attribute partition defined by a request to the contents of the portions of the data structures defined by portions 203, 204 and 205, respectively, of the dispatch specification. If the respective portions of the verb match the contents of the corresponding portions of the data structures (Fig. 8B), the kernel 13, 14 initiates the procedure defined in the dispatch entry 134, which is taken from the portion 206 of the dispatch specification (Fig. 3E)

B. DATA FILES AND USE

1. Data Dictionary

When a management module is enrolled, its management specification may define new global entity classes, subentity classes or attributes, directives or events of global or subentities. The management specification (Figs. 3A through 3D) is used to construct a data dictionary, which, in turn, is used in constructing other data structures, which are described below in connection with Figs. 5, 8A and used as depicted in Fig. 9. The data dictionary comprises a hierarchical database having the general schema or structure shown in Fig. 4. With reference to Fig. 4, the schema has a relative root node 220 which is associated with a global entity as defined in the management specification (Fig. 3A). The global entity node points to a plurality of subsidiary nodes in the hierarchical schema, including a subsidiary node 221 listing all attributes, subsidiary node 219 listing attribute partitions, a subsidiary node 222 listing attribute aggregations, a subsidiary node 223 listing directives, and a subsidiary node 224 listing subentities, of the entity body 53 in the entity definition 46 of the management specification.

Each of the subsidiary nodes 219 through 224, in turn, points to the respective elements defined in the entity body. That is, the attribute node 221 points to attribute definition nodes 225 each of which contains the definition of an attribute defined in an attribute definition 54 in the entity body 53, the attribute partition node 219 points to attribute partition nodes each of which contains an attribute partition defined in a partition definition 56 in the attribute definition 54 of the entity body 53, the aggregations node 222 points to aggregation definition nodes 226 each containing the definition of an aggregation defined in an aggregation definition 55 in the entity body 53, the directives node 223 points to directive definition nodes 227 each containing the definition of a directives defined in directive definition 56 in the entity body 53, and the subentities node 224 points to subentities definition nodes 228 each containing the definition of a subentity defined in a subentity definition 57 in the entity body 53. Each of the directive nodes 227, in turn, points to a request node 230, a response node 231 and an exception node 232, each of which, in turn, contains the definition of a request, response and exception as taken from the request definition 90, response definition 91 and exception definition 92 (Fig. 3C) of the management specification. In addition, each subentity node 228 forms the root node of a subschema having a structure similar to that depicted for a global entity shown in Fig. 4, including a subsidiary node 233 for attributes, a subsidiary node 234 for aggregations, a subsidiary node 235 for directives, a subsidiary node 237 for partitions and a subsidiary node 236 for subentities. The schema depicted in Fig. 4 is repeated for all subentities and their subentities as defined in the management specification depicted in Figs. 3A through 3D.

The information in the management specification is merged into the respective nodes of the data dictionary and is used to create the user interface information file 29 used by a presentation module 10 in connection with display of entity information, including entity identification information and response information, to an operator and generation of requests for processing by the other portions of the control arrangement and the entities of the complex system, as described below. The diverse nodes of the data dictionary receive the information from the elements of the management specification to form the complete database comprising the data dictionary. The information in the dispatch specification (Fig. 3E) is used to create the dispatch tables 28, as described below in connection with Figs. 8B and 9.

With this background, Fig. 5 depicts a single presentation module 10, functional module 11 and access module 12, the kernel 13, 14 including information manager 15, 20 and dispatcher 16, 21. In addition, Fig. 5 depicts various portions of the data storage element 17, 22. Specifically, the data storage element 17, 22 includes a configuration and domains database 23, 25, an alarms database 24, a historical data file 26, a data dictionary 27 and a dispatch table 28.

2. Historical Data File

The historical data file 26 contains information regarding the status and condition of entities, in the case of the presentation-functional aspect of kernel 13, and entities, in the case of the functional-access aspect of kernel 14. In file 26 the status and condition information also includes timing information, to identify the time at which the status and condition information was generated. When the information manager 15, 20 receives a request, or a subordinate request, regarding status or condition at a specific time, it determines whether the information is in the file 26, if the time indicated in the request or subordinate request is in the past, and responds using the contents of the file.

On the other hand, if the time indicated in the request or subordinate request is a future time, the information manager 15, 20 effectively schedules the request to be processed at the time indicated. That is, the information manager retains the request or subordinate request until the indicated time is reached, and at that point processes the request either using responses directly from the access module 12 or functional module 11, as appropriate, or using the contents of file 26 as appropriate.

These functions will be fully described below under the heading "Scheduling".

3. Dispatch Table

The dispatch table 28 is used by dispatcher 16, 21 to determine how to transfer a request or subordinate request to the appropriate functional module 11 or access module 12. The contents of the dispatch table 28 identify the locations, in the distributed digital data processing system, of the entry points of the routines comprising each of the functional modules 11 which may be called in response to requests from a presentation module 10. More specifically, the dispatch table 28 contains calling information which facilitate initiation of the various operations by the respective functional modules 11. Similarly, the contents of the dispatch table 28 identify the locations, in the distributed digital data processing system, of the entry points of the routines in the access modules 12 which are used to process subordinate requests from a functional module 11, that is, the calling information defining the various operations by the respective entities.

4. User Interface Information

The control arrangement further includes a user interface information file 29 that contains information as to the various functions provided by the functional modules 11 and the entities controlled by the access modules 12. The user interface information file 29 contains information derived from the management specifications of the respective entities. The presentation modules 10 use the contents of the user interface information file 29 in displaying menus and other objects on the operators' terminals to facilitate control of the complex system. The information in the user interface information file 29 facilitates display of the various functions and operations in connection with the complex system's entities.

5. Configuration Database

As discussed above, a configuration functional module may create and maintain a configuration database, which lists all of the entity instances in the current configuration of the complex system (and also past configurations, if desired). This information may be used, e.g., by a presentation module to create parse tables or user menus listing available entity instances. The configuration database may also include a domain database for limiting the scope of control of a user, to facilitate use of the complex system, as discussed below.

In addition to the above features, in one embodiment, the configuration database may be used in conjunction with presentation modules to support wildcarding in user commands. When a user command containing a wildcard is received by a presentation module, the presentation module issues a request to the configuration functional module, requesting an enumeration of all entities in the configuration that match the wildcard request. The configuration functional modules then uses the information in the configuration database (along with domain information) to produce the list. After receiving the list, the presentation module expands the user request into all of the possible subsidiary requests which match the wildcarding.

For example, the request

SHOW NODE * IN DOMAIN SITE1

(where SHOW is the directive, DOMAIN is the domain entity class, SITE1 is a domain instance, NODE is a global entity class, and * is the wildcard) would be interpreted as a command to show all instances of nodes within the domain named SITE1. The presentation module would thus expand the request into several requests, each of the form

SHOW NODE ⟨instance⟩

(where ⟨instance⟩is the instance name), one corresponding to each instance of the NODE class in domain SITE1.

Partial Wildcarding may also be supported. In this case, the group of target entities with instance names that match the pattern specified by the partial wildcarded name are issued directives. For example, "NODE *OO" would match "NODE FOO" and "NODE MAGOO", but not "NODE BAR". Partial wildcarding may not be used in fields having identifiers with certain datatypes, e.g., identifiers which do not use text or digit strings.

In preferred embodiments, wildcard expansion is not allowed in the global entity class field of a user directive. Global class specifications are not wildcarded because doing so would result in insufficient control on the scope of a command. This may create errors if directive names supported by one entity class are not supported by another. Even where a directive name is supported by multiple classes, the directive name may correspond to unrelated functions in different classes, causing undesired side-effects (e.g. a "DELETE *" directive). In addition, global entity wildcarding may simply produce more information than the user intends (e.g. a "SHOW *" directive). Note that wildcarding may be safely allowed in subentity classes.

Embodiments of wildcarding may also delegate some or all of the wildcard expansion duties to access modules. This is particularly the case where no configuration functional module is used. In the absence of a configuration functional module, the access modules (ordinarily associated with accessing all modules of a class or subclass) may store instance data as part of their private storage 32 (Fig. 2B). In this case, the access modules would use the instance data to expand wildcards in received requests. If wildcarding is not supported by a particular access module, an exception indicating this condition would be returned to the user.

C. DATA FILE MANAGEMENT AND ENROLLMENT

When a management module is added to the control arrangement, or when new information relating to management of the entities becomes available, the control arrangement must adapt. The control arrangement is data driven, and thus adapting the system to new modules or information involves modification of the relevant data files. In general, this process is known as data file management. The particular procedure by which the control arrangement adapts to a new module is known as enrollment.

1. Historical Data File Management

In one specific embodiment, the contents of the historical data file 26 are provided and maintained in part by a functional module 11 which serves as a historical data recorder functional module. In that embodiment, the historical data recorder functional module is controlled by an operator through requests presented to a presentation module 10. Initially, a such request, which identifies an entity and one or more attributes, along with a polling rate, establishes a record in the historical data file for the identified entity and attributes and enables the historical data recorder functional module to issue, at the polling rate specified in the request, subordinate requests to the entity enabling it to respond with value(s) representing the condition(s) of the entities of the complex system specified by the entity and attribute(s) specified in the request. In addition, other types of requests permit an operator to initiate other operations in connection with the historical data recorder functional module, including changing the polling rate, temporarily enabling and disabling the polling, and showing the last value in a response.

2. Dispatch Table

The contents of the dispatch table 28 and of the user interface information file 29, comprise enrollment information, and are provided by the various functional modules 11 and access modules 12 during an enrollment procedure. During an enrollment procedure in which a module enrolls in the control arrangement, it loads the display information, including name and code information from its name fields into the data dictionary. In addition, the module loads the code information and other information as defined by the management specification from the data dictionary (Fig. 4), and the dispatch information from its dispatch specification (Fig. 3E) into the dispatch table 24.

3. User Interface Information

The presentation modules 10 use the display information in the user interface information file 29 to determine, first, whether to display an entity, attribute, directive, and so forth, and, second, what to display. The user interface information file 29 forms a parse table that, in response to a command by an operator at a terminal, enables the presentation module 10 receiving the command to parse the command using the parse table to derive codes, corresponding to the codes for the request, entity and attributes defined in a management specification, which it transmits as a request to the kernel 13.

Note that functional and access modules do not need to have any user interface code. All user interface support is provided to these modules, and the module designer need not concern himself with the user interface. This simplifies module design tremendously, and guarantees that the system will have a uniform look and feel to the user, regardless of the actual modules in use.

Upon receiving a request from a presentation module, the dispatcher 16 calls the functional module 11 using the dispatch information in the dispatch table 28. The dispatch table 28 also forms a parse table, which the dispatcher 16 uses to dispatch to the proper procedure to process the request, as described below in connection with Fig. 9.

It will be appreciated that the use of codes in the parse table and in the dispatch table 28, while presentation specific information is being used in the user interface information file 29, essentially separates the identifications of the entities, attributes, and so forth, as used by the dispatcher 16, from the identifications displayed to the operators by the presentation modules 10. The display generated by the presentation modules 10 may, therefore, be in diverse languages, while the requests generated by the presentation modules 10 contain the same identifications of the entities, attributes, and so forth.

In addition, the user interface information file 29 may store information which is already available from the configuration database and data dictionary in a more convenient format.

For example, the class data in the data dictionary (Fig. 4) indicates all of the directives 223 supported by entities in the complex system. However, the directives 223 are stored in a hierarchical format, and are subordinate to the entity classes 220. Although this format is logical for representing entity class information, it is less useful for a parse table. A user request typically lists the directive first (e.g. "SHOW" in "SHOW NODE FOO"), thus a parse table should have directives as the first level of a hierarchical structure. As can also be seen by the above example, a parse table may need to parse a command where class names (e.g. "NODE") are mixed with instance names (e.g. the identifier FOO in "NODE FOO"). Therefore, after a listing of the available directives, the parse table should list the class names which support those directives, and then the data types of instances of those classes. Although the class and data type information is available from a reorganization of the Data Dictionary, for expansion of wildcards, instance data can be obtained from the Configuration Database. Thus the parse tables in the user interface information file can consolidate directive and entity class, making the parsing of user input computationally more efficient.

The above example also applies to a graphical or menu-driven interface. However, in this type of interface, the user may wish to set a context for his commands, by graphically selecting a particular entity or domain of entities for the subsequent operations, and the OSI category (as listed in the category field 87 of the directive definitions) of the directives to be made. Next, a menu could be generated which listed all of the supported directives. The user could request a directive for one or more instances (e.g., by clicking on the directive and instance) or an entire domain or entity class (e.g., by clicking on the directive alone) using the pre-formed menus. On a EXAMINE or CATEGORY type directive, further menus may prompt the user to select attribute partitions or aggregations.

To implement this type of interface, a listing of all of the domain and entity instances and a listing of all of the instances in a domain must be fetched from the configuration database. In addition, a forms database may store the directives supported by the class or domain.

The user interface information file may also store default value information. Default values for instances or classes may be provided by the user or by the Management Specification for the relevant entity class. This allows the user to save typing time by specifying a default value in a command. For example, the user may be most concerned with NODE FOO, and may specify "NODE FOO" as the default node. Later, the user can type a command such as "SHOW ROUTING", which would be interpreted as "SHOW NODE FOO ROUTING". Similar uses of default values can be used in a graphical environment.

Another example of user interface information is an on-line help file which is available to the user through presentation modules. The help file contains help information for using the existing set of management modules. In preferred embodiments, the help file is constructed from help information supplied by the modules when they are enrolled. The supplied help information may include a text description of the entity and subentity classes supported by the module, and the directives to those classes supported by the module. In addition, tutorial information can be supplied to educate a first-time user on the use of the module and its directives. The above information may also be determined from the management specification for the module, however, the help information file translates the management specification information into english sentences, reducing the need for a user to learn the syntax of the management specification.

4. Historical Data Recorder

The historical data recorder functional module 11 may use the entities' polling information from its portion of the data dictionary, including the portions relating to the maximum polling rate field and the default polling rate field, to initiate and control polling in connection with the entity's various attributes as defined in the attribute definitions 54, the responses to which the historical data recorder functional module 11 stores in its historical data file 26.

5. Module Enrollment

With reference to Fig. 5, An access module 12, for example, while it is engaged in an enrollment procedure, loads display information, including the name and code information defined in the name and code information from its name fields and information from the portion of its data dictionary related to the display fields in its management specification into the user interface information file 29. Similarly, a functional module 11 loads the code information and other information as defined by the management specification from the data dictionary (Fig. 4), and the dispatch information from the dispatch specification (Fig. 3E) into the dispatch table 28.

D. INTERMODULE AND INTER-NODAL COMMUNICATIONS

1. Control Functional Module

In one specific embodiment the operator may control an access module 12 directly, through a control functional module 11 that essentially generates subsidiary requests which are copies of requests which it receives from the dispatcher 16. In that embodiment, the presentation module 10 that receives the command, parses the command using the parse table in the user interface information file 29 to derive codes corresponding to the codes for the request, entity and attributes of the access module 12 defined in a management specification, which it transmits as a request to the presentation-functional kernel 13. The control functional module 11 passes the request as a subsidiary request to the functional-access kernel 14, where it is treated in the same manner as any other subsidiary request.

Upon receipt of a subsidiary request from a functional module 11, the dispatcher 21 calls the access module 12 using the dispatch information in the dispatch table 28. The dispatch table 28 also forms a parse table, which the dispatcher 21 uses to dispatch to the proper procedure to process the request, as described below in connection with Figs. 9A and 9B.

2. Inter-Nodal Communications

If the control arrangement controls a complex system comprising a distributed digital data processing system, Fig. 5 generally depicts elements, including a presentation module 10, a functional module 11 and an access module 12, including kernel 13, 14 comprising information manager 15, 20 and dispatcher 16, 21 and associated data files 23, 24, 25, 26, 27, dispatch table 28, user interface information file 29, all included in a single process in a single node of a distributed digital data processing system. If the distributed digital data processing system includes a presentation module 10, a functional module 11 and an access module 12 in different processes or nodes, the control arrangement includes a dispatcher 16, 21 in all processes and nodes. With reference to Fig. 6, when a dispatcher 16(1) in one process in a node receives a request from a presentation module 10(1) which must be processed by a functional module 11(2) in a second process or node, it transmits the request, by an interprocess communication mechanism, if the functional module 11(2) is in another process on the same node, or an internode communication mechanism to a process on the other node, to a dispatcher 16(2) in the other process or node. The dispatcher 16(2) then selects a functional module 11(2) to process the request. The dispatcher 16(2) receives the response generated by the functional module 11(2) and transmits it, by means of the interprocess communication mechanism or internode communication mechanism, to the dispatcher 16(1), which, in turn, enables a presentation module 10(1) to display the response to the operator.

Similarly, when a dispatcher 21(2) receives a subsidiary request from a functional module 11(2) to be processed by an access module 12(3) in another process or node, it transmits the subsidiary request to a dispatcher 21(3) in the other process or node by means of the interprocess communications mechanism or internode communication mechanism, respectively. The dispatcher 21(3) then transmits the subsidiary request to the access module 12(3) for processing. The dispatcher 21(3) receives the response from the access module 12(3) and transmits it, by means of the interprocess communication mechanism or internode communication mechanism, to the dispatcher 21(2), which, in turn, couples it to the functional module 11(2).

3. Request and Subsidiary Request Structure

The structure of a request, and specifically the parameters that are included with the request, is depicted in Fig. 7A. The structure and contents of dispatch table 24 (which are similar to the structure and contents of dispatch table 26) will be described in connection with Figs. 8A and 8B. Thereafter, the process performed by the information manager 15, 20 and dispatcher 16, 21 in connection with parsing of a request will be described in connection with Fig. 9.

With reference to Fig. 7A, a request, which may be generated by a presentation module 10 in response to operations by an operator in connection with the contents of user interface information file 27, or which may be generated by information manager 15 during polling in connection with the various entities of the complex system being controlled, includes a plurality of parameters. All requests have the same structure, including an initial call identification, which is not shown, followed by parameters, which are depicted in Fig. 7A. As discussed above, the kernel 13, 14 has a single dispatcher 16, 21 having a presentation-functional aspect 16 and a functional-access aspect 21. Which of these aspects are respectively enabled by a request is determined by the initial call identifier. The initial call identifier may indicate a call to a functional module or an access module, and is respectively routed to the corresponding aspect of the dispatcher. A presentation or functional module may call a functional module, and a functional module or access module may call an access module, but a presentation module may only call an access module through a "control" functional module, as discussed above.

The parameters include a verb field 120 whose contents identify the type of request, that is, an operation to be performed in processing the request. As noted above, a request may cause a functional module 11 or access module 12 to initiate a change in the status or condition of an entity in the complex system being controlled, it may initiate a return of information as to the status or condition of such an entity, or both. The contents of verb field 120 indicates the operation to be performed by the functional module 11 or access module 12.

In addition, a request includes an input entity specification field 121, which identifies the entity in the complex system being controlled. If the verb is a non-action verb, for example, if it requests a response indicating the values of one or more attributes, the request includes an attribute pointer field 122 which contains a pointer to one or more attributes in connection with which the operation, defined by the verb and entity class, is to be performed. If the verb is an action verb, that is, if it causes a change in the specified entity, the request does not have an attribute pointer field 122.

In addition, a request includes an input time specifier field 123 that contains a pointer which points to a time data structure that contains certain timing information, including the absolute system time, time interval definition, and the time accuracy specification, and an indication as to the time range of interest in the request, for scheduling purposes. An input/output context handle field 124 contains a value which identifies the request in the context of a multiple-part operation, each part of which requires a separate request. An output entity specifier field 126 contains a pointer to a data buffer which can be used by the dispatcher 15 (or dispatcher 21, if the parameters form part of a subsidiary request) in connection with identification of the entity.

A request also includes an output time specification field 126 that contains a pointer to a time stamp specification which is to be used by the functional module 11 (or access module 12 in the case of a subsidiary request) in connection with formation of the response. Finally, an optional data descriptor field 127 contains descriptors to buffers containing data which is to be used in processing the request and in which the entity is to store data comprising a response, respectively. Each descriptor includes a pointer to the starting location of the respective buffer and a length specifier indicating the length of the buffer.

In alternative embodiments of the invention, the request may also include qualifier fields, as a separate parameter or as an additional element of the parameter fields discussed above.

A WITH qualifier can be associated with the Entity field to, for example, filter the entity list produced by a wildcard. For example, "BRIDGE * WITH STATUS = 'ON' AND FILTERING = 'OFF'" refers to every bridge class entity with its status flag set to ON and filtering flag set to OFF. (This example also illustrates the use of boolean functions such as AND, OR, NOT and XOR with qualifiers.) In preferred embodiments, to implement the WITH qualifier, all modules and the information manager are configured to check for the presence of a WITH clause at each level (i.e. global entity, subentity, sub-sub-entity) of the Entity parameter.

Other qualifiers may be used as a distinct parameter of the request. For example, communications qualifiers include: a "TO ⟨filename⟩" qualifier which sends the response of a request to a file named ⟨filename⟩; a "FROM ⟨filename⟩" qualifier which retrieves other request parameters from a file named ⟨filename⟩; a "VIA PATH" qualifier which specifies a series of "hops" along a path, through a hierarchy of management modules (useful in specifying, e.g., the precise management module among several arrangements that will perform the operation); and a "VIA PORT" qualifier which specifies a particular network path a management module uses when performing the operation (useful, e.g., to specify that an access module will perform a diagnostic test using a specific EtherNet port.)

Similarly, distinct parameter qualifiers may specify a group of entities of interest. The "IN DOMAIN ⟨domain name⟩ " qualifier filters the directive to apply only to members of the domain named ⟨domain name⟩.

Also, distinct parameter qualifiers may authenticate or authorize the requestor of management services which have limited access privileges. The "BY ACCOUNT", "BY PASSWORD", and "BY USER" qualifiers are examples which specify the account name, password, or user ID of the requestor for these purposes.

In addition to the above, qualifiers specify the time that a directive should be executed. Generally, this is accomplished with an AT clause. For a show command, the syntax of an AT clause is:

⟨AT-clause⟩::= "AT" ⟨time-arg⟩{"," ⟨time-arg⟩}

where the time argument ⟨time-arg⟩may, e.g., indicate the start time ("START = ⟨time⟩"), the end time ("END = ⟨time⟩") or duration ("DURATION = ⟨time-length⟩"), the period of repetition ("REPEAT EVERY [=] ⟨time-length⟩"), the time accuracy ("CONFIDENCE [=] ⟨time-length⟩), or the sampling rate ("SAMPLE RATE [=] ⟨time-length⟩). These arguments may interact with one another to create a general schedule and scope of interest for a request. In particular, in one particular embodiment, the three time arguments, START, END and DURATION are related such that any two of them define a period. Thus when a time-normalized entity statistic is displayed, at least two of these qualifier arguments must be specified.

Other time qualifiers may also be used. For example, a time qualifier of AT OR BEFORE ⟨time⟩can be interpreted as a request for any information with a time stamp at or before the time given by ⟨time⟩. Upon receiving a request with such a qualifier, an management module will continuously check for actions which produce the requested information. If the information is produced, for example by the actions of another party, it will be returned to the requestor. Otherwise, the management module will continue to check for the information until time ⟨time⟩arrives. If the information is produced, then it will be returned to the requestor. Otherwise, at time ⟨time⟩, the management module will force a poll of the information from access modules or the entities, and return the information to the requestor.

To complement the AT OR BEFORE time qualifier, a NOW time qualifier can also be implemented. This qualifier would immediately force a poll of the requested information.

E. TIME

As discussed above, the request structure includes a time specifier field 123. In addition, a field 124 contains a handle pointer to a context data structure, which is a dedicated segment of memory for storing processing context information. The handle is used as a "notepad" for communication of, for example, context information between modules and the information manager.

1. Timestamps

Each item of data contains a timestamp value. In the case of data returned to the user or a management module, the timestamp indicates: the instant of time at which an event described by a data item happened, the instant of time that applies to the data value(s) returned for a directive, or the instant of time when a requested action was actually performed. In the case of historical data stored in the historical data file, the timestamp indicates the instant of time at which a given data item had a particular value. For the purposes of the historical data file, a timestamp can be considered as a key or index. A scope of interest time specification 123 may be used to request the retrieval of a particular piece of stored information with a given key or index.

2. Scope of Interest

Scope of interest time specifications are supplied by requests using the time specifier field 123. Using a time specifier, other values of data than "the value it has right now" can be displayed and processed, and statistics can be computed over some time period. In one particular embodiment, a time "scope of interest" is expressed by prepositional phrases in the time specifier of a request. Generally, a time specifier is used with a SHOW command, hut time contexts may also apply to MODIFY type requests and actions.

Time scopes of interest can be indicated by either an absolute instant, a sequence of absolute instants, an interval (start time "START" and duration "DUR"), a repetition of instants, or a repetition of an interval.

Any of these may have associated with them a relative time period ("EVERY") that specifies the periodicity with which the instant, instants, or interval is repeated. When a period is specified, the original instant, or sequence of instants or interval is treated as a base, to which the period is added, repetitively. For example, the time specification "5:00 EVERY 0:15" is equivalent to 5:00, 5:15, 5:30, 5:45, ... An absolute time instant ("UNTIL") can be specified to indicate when the repetition is to terminate. For example, the time specification "5:00 EVERY 0:15 UNTIL 6:00" is equivalent to 5:00, 5:15, 5:30, 5:45, 6:00. Repeating intervals may be specified in the same way. "START 5:00 DUR :05 EVERY 1:00" is equivalent to the intervals 5:00-5:05, 6:00-6:05, 7:00-7:05, ...

3. Scheduling

Scheduling information is also provided by time specifier field 123. Specific scheduling times can be indicated by either an absolute instant, or a sequence of absolute instants. Unlike scopes of interest, scheduling times may not include an interval. Intervals whose begin and end points are equal resolve into instants (e.g. (TODAY,TODAY)).

A few rules apply to intervals. Intervals in the past may have begin points denoted by the keyword YESTERDAY, or an absolute time in the past. Similarly, intervals in the future may have a begin point denoted by the keyword TOMORROW, or an absolute time in the future. Also, the start time of an interval must be earlier than its end time.

4. Time Context Handle Structure

As discussed above, the scheduling and scope of interest information may be supplemented in a request with an associated context handle. The handle is created by the module which executes the request, and is subsequently used in communication with the service provider. When a call is received by the service provider, e.g. the Information Manager, a context block is created as a local reference to the request's time context.

Generally, context blocks and handles are used as references to the status of a request. As the initial request can generate many subsidiary requests, it is possible that many handles and context blocks can be created by a single request. The context blocks are the reference used by a service provider, whereas the handles are the reference used by the service requestor. Each process (i.e. module or information manager) in a request/subsidiary request chain knows only about the context block and handles relating to its local part of the chain.

Referring to Fig. 7B, in one particular embodiment, a time context handle 172 created by a requestor, e.g., a presentation module 10, includes a scope field 175 and schedule field 176 which relate to the time specification 123 of the initial request. These fields supplement the data in the time specifier of the request, and are used to determine the current status where multiple requests and responses exist for a single operation. The handle 172 also includes a context pointer 177 and a state variable 178. These data items provide the status and reference functions of the handle, and are created and stored with the scope and schedule fields 175, 176 when the request is made.

Where multiple requests and responses exist for a single operation, the context field 177 will eventually contain a pointer to an additional data structure 174, known as a context block, which is created and maintained by the service provider, e.g., the presentation-functional aspect 15 of the Information Manager (functional or access modules may also create and maintain context blocks in response to requests), in response to an initial request requiring multiple responses.

The state field 178 of the handle contains one of three values: "FIRST", "MORE", or "CANCEL" which are used as flags to indicate further actions that should be undertaken. When first created, the handle state is set to "FIRST".

As discussed above, if a request can be satisfied by a single response, the response is generated and returned to the requestor. In the more general case, the service provider, e.g., a functional module, information manager, or an access module, cannot satisfy the request in one reply. For example, the requestor may have used wildcarding in the input entity parameter 121, to specify a group of entities. As each reply can only incorporate information from a single entity, several replies are required, one for each entity. In another case, a request to a single entity may have a time specifier with several different time values. As each reply can only incorporate information for a single time value, several replies are required, one for each time. A request that requires multiple replies can be for any type of operation, including obtaining attribute data about an entity or entities, modifying attributes of several entities, and modifying the state of several entities.

When the service provider processes the request and determines that it has additional replies, it is responsible for indicating this to the requestor. Thenceforth, the requestor is responsible for querying the service provider for the additional replies. To implement this, intermediary processes, e.g., the Information Manager, must save the information relevant to the request that it has generated.

The latter function is accomplished by creating a context block 174, which may contain relevant private variables 173 that have been generated in responding to the request, such as a pointer to the dispatch entry of the service provider (see discussion under Dispatch Table, below), as well as a context pointer(s) 179 to any handles that relate to subsidiary requests to, e.g. a functional module.

The handles and context blocks are used as follows. The service provider notifies the requestor that it has additional replies by using the appropriate handle modification routines to: (1) save a pointer 177 to its context block 174 in the requestor's handle 172, and (2) set the state field 178 in the requestor's handle 172 to a value of "MORE". When the reply is returned to the requestor, the requestor sees the "MORE" state in its handle state field and thus knows that the service provider has additional replies for this request. If the requestor does not want these additional replies, it must cancel the request (see below). If the requestor wants the additional replies, the request must be repeated, without changing any parameters.

When the service provider receives these repeated requests (which will have a handle state field 178 equal to "MORE"), it searches for and detects the "MORE" state using the appropriate handle access routine. Then the service provider knows that the calls are part of a previously established request. (Note that a handle with a state of "FIRST" indicates to the service provider that the associated call is the first call of the request.) For each call with a "MORE" handle state, the service provider retrieves the context block 174 pointed to by the handle context field 177, and uses the context block to continue its execution to provide the additional replies. There is only a single reply for each call made to the service provider. As long as the service provider maintains the handle parameter in the "MORE" state, it has more replies for the request.

When the service provider is returning to the requestor with its last reply (determined by, e.g., the scope and schedule fields 175, 176 in the requestor's handle), the requestor's handle state field 178 is set back to a value of "FIRST" (the initialized state). When the return is made to the requestor with this last reply, the requestor sees its handle parameter state set to "FIRST" and knows that its request has been fully satisfied. Note that if the request is satisfied with a single reply, the service provider retains no context and never causes the state of the handle parameter to become the "MORE" state. The requestor's handle stays at its initialized "FIRST" state, indicating to the requestor that the request is completed.

When a service provider returns the handle parameter in a "MORE" state, the request must be repeated or cancelled. If the request is otherwise abandoned, system resources will be lost, owing to the memory allocated to the handle and context block.

Note that for the above discussion, if the service provider did not issue subsidiary requests, a single handle would suffice for communications between the service requestor and provider. However, if the service provider did issue subsidiary requests there would be more than one separate handle--the initial requestor's handle, which is provided by the requestor for the call, and different handles created by, e.g., the Information Manager and forwarded to, e.g., an access module.

Where multiple requests and responses exist for a single operation, scheduling subsidiary requests to the service provider is performed by the Information Manager, and is controlled by the schedule time component of the time specification parameter 123. For each schedule time specified in the time specification, the Information Manager will create a request which causes the service provider to perform the requested operation and issue responses.

When the service provider has completed the requested operation, it issues a response. When the Information Manager sees that the service provider has completed the requested operation, it then examines the schedule time context that it keeps for the initial request. If there are further times for which the requested operation is scheduled, the Information Manager does not set the requestor's handle state to "FIRST", but leaves it in the "MORE" state. The requestor sees its handle parameter still in the "MORE" state, and knows that the full request has not been completed, and asks for the remainder. The Information Manager then causes a wait until the specified schedule time, then allows the Dispatcher to perform another call to the service provider. Note that the service provider cannot distinguish this next call from that of a completely new request, as it has retained no context after returning with its handle state set to "FIRST". Also, the requestor does not distinguish between a handle state of "MORE" caused by the service provider having more replies to a request and the Information Manager preparing for a new schedule time instant.

In other embodiments, the handle access routines would be enhanced to permit the client to determine the cause of the "MORE" state of the handle parameter.

If, during a request with multiple replies or multiple schedule times, the requestor decides that it does not want any further replies from the service provider for this request, it must cancel the request. Possible reasons for wishing to cancel the request include receiving an exception reply that indicates that further data will not be useful, or receiving an error condition that indicates that the desired operation is not performing properly. The reasons for canceling are the responsibility of the requestor. A cancel terminates all activities of the request, including any scheduling and scope of interest operations.

A cancel can be done when the service provider returns to the requestor with a handle parameter state of "MORE". The requestor performs the cancel by using the appropriate handle modification routine to change the handle parameter state to a value of "CANCEL" and re-issuing the call. The requestor must not change any other parameters for this call. When the service provider receives this call, it sees the handle parameter in a state of "CANCEL" instead of the expected "MORE" state. It retrieves its context from the handle parameter and uses that context to perform any cleanup required. This cleanup includes canceling lower level requests that it is making, terminating any processing, and returning any system resources. When the service provider has completed its cleanup, it uses the appropriate handle modification routine to re-initialize the handle parameter back to the "FIRST" state. It then returns with the special condition value return code of CANCELED to indicate a successful cancel of the request.

The requestor cannot cancel a request after the service provider has returned with a handle parameter state of "FIRST". The request is already completed, and no service provider context exists to cancel. Therefore, the cancel routine described above will return an error if the handle state is not "MORE".

F. DISPATCH

The dispatch tables 28 include a plurality of data structures, one of which is shown in Fig. 8A, and one or more dispatch lists including dispatch entries, one of which is depicted in Fig. 8B. The dispatch tree and dispatch lists essentially form parse tables which are used in parsing a request, as described below in connection with Fig. 9. With reference to Fig. 8A, a dispatch tree includes a plurality of entity nodes 130. The entity nodes 130 are organized in a tree structure to assist in parsing, but they may be organized into other data structures. The entity nodes identify the various entities in the complex system in connection with which a request may be issued. The entity nodes 130 include pointers which point to dispatch entries 134 (Fig. 8B) in the dispatch lists maintained in respective dispatch tables 28.

The term "entity node" is used to describe the data structure 130 because it satisfies the entity model set forth above. Generally, data structure 130 satisfies the entity model because it has a hierarchical structure and its child structures resemble it. The term "entity node" as it is used to describe data structure 130 should not be confused with the term "entity" used to describe elements of the complex system.

An entity node 130 includes several fields, including a class/instance flag field 140 which indicates whether the entity node 130 relates to an entity class or an instance within a class. Each entity may be an instance of a class, the class being defined by a class name identified in the entity's entity definition 46 (Fig. 3A), and the dispatch table 24 includes separate entity nodes 130 associated with the class and the instance, as described below in connection with Fig. 9.

While parsing a request, the class names and instance names of an entity and its subentities are parsed using data structures of the type shown in Fig. 8A, although the structure is used differently in parsing the class names or instance names. The class or instance case is indicated by the class/instance flag.

The entity node 130 also contains tree link pointers that identify various other elements in the dispatch table 28. A module which services requests relating to several entities of the same class may be identified by means of a wildcard or an ellipsis. If so, an entity node associated therewith has a wildcard pointer in a field 141 or an ellipsis pointer in a field 142. Each wildcard pointer and ellipsis pointer comprises a tree link entry, as described below. If the entity node relates to a class which has no instances, an example of which is described below in connection with Fig. 9, a field 143 contains a null pointer comprising a tree link entry to another entity node. Finally, a field 131 contains a coded entry, which contains the code identifying the class or the name of an instance of the entity associated with the entity node as well as a link pointer.

The coded entry field 131, depicted in the entity node 130 on Fig. 8A, is one entry in a coded list. (The remainder of the list is not shown.) The coded list is a linked list which contains names of classes of entities defined by the management specifications of the entities (see Figs. 3A through 3D), when referring to classes or names of instances of entities. Each coded entry 131 includes a pointer 150 to the next coded entry in the list, a class code/instance name value field 151, and a field 152 which contains a links entry 133 which includes a pointer to an entity node 130 or to a dispatch entry 134.

The class code/instance value field 151 in the coded entry 131 contains either a class code or an instance name. The contents of field 151 comprise a class code if the class/instance flag field 140 of the entity node 130 is conditioned to identify the entity node as being related to a class. Alternatively, the contents of field 151 comprise an instance name if the class/instance flag field 140 of the entity node 130 is conditioned to identify the entity node as being related to an instance.

Referring to Fig. 8B, the dispatch entries 134 in a dispatch list are used to identify the particular procedure to process a request. A dispatch list is a linked list of one or more dispatch entries 134, each entry 134 containing information useful in transferring a request or subsidiary request to an appropriate functional module 11 or access module 12. More specifically, a dispatch entry 134 includes a pointer 160 to a next dispatch entry 134 in the list. A field 161 includes an identification of the functional module 11 or access module 12 during whose enrollment the dispatch entry 134 was generated. A dispatch entry 134 also includes a series of fields 162 through 164 which point to a procedure, process and node in the complex system for processing a request. A field 165 identifies the verb with which the dispatch entry is associated and an attribute field 166 identifies a set of attributes, as identified by attributes defined by an attribute definition field 54 of the management specification (Fig. 3B). Finally, a count field 167 identifies the number of times the dispatcher has used the dispatch entry 134 in connection with processing a request or subsidiary request.

With this background, the process performed by dispatcher 16 in parsing and dispatching a request from a presentation module 10 will be described in connection with Fig. 9. It will be appreciated that the dispatcher 21 performs a similar process in connection with a subsidiary request from a functional module 11. With reference to Fig. 9, a request 180 as follows:

```
SHOW
    NODE <node name>
        ROUTING
            CIRCUIT <routing circuit name>
                CHARACTERISTICS
```

which is used in connection with a distributed digital data processing system. The request 180 includes a number of sections, including a verb section 181, namely SHOW, an entity section comprising a plurality of entity class codes and instance names 182 through 186, and an attribute section 187 comprising a plurality of attributes. In this example, the verb SHOW initiates generation of a response from the entity named in the request, relating to the named characteristics.

In the request 180, the entity section, namely, elements 182 through 186, includes a number of class/instance pairs. In particular, element 182, NODE, is a class code, and element 183, namely, ⟨node name⟩ identifies, by instance name ⟨node name⟩an instance of the entity class NODE. In the distributed digital data processing system, ⟨node name⟩ identifies a node in the distributed digital data processing system.

In addition, the request 180 further includes, in the entity section, an entity class code 184, ROUTING, which has no instances. In addition, the request 180 has a further entity class code, CIRCUIT, which has an instance identified by ⟨ROUTING CIRCUIT NAME⟩

With reference to Figs. 3A through 3D, which depict a management specification, various elements of a request in connection with an entity are specified by the management specification. Specifically, the contents of the verb section 181 of a request are taken from the directives defined by the directive definitions 56, the entity class and sub-entity class names 182, 184, 185 are taken from the entity class code field 47, and the attributes section 187 is taken from the attribute definitions 54 of the management specification for the entity.

The entity and sub-entity instance names are taken from instance data known to the user (for example, by reference to the configuration database or through automatically generated menus).

In response to the receipt of a request, the dispatcher 16 first begins parsing the request in the entity section, beginning with global entity class code element 182, using entity nodes 130 (Fig. 8A). In particular, with reference to Fig. 9A, the dispatcher 16 first (step 190) begins at a root entity node 130, which has a class/instance flag 140 which identifies the entity node as being associated with class codes, and searches for an entry of its coded list 131 which contains a coded entry 131 that, in turn, has a class code field 151 which contains a class code of NODE. If the dispatcher 16 is unable to find such an entry in the dispatch table 28, it searches for a wildcard or ellipsis pointer (see below). (If no wildcard or ellipsis pointers are found, it responds with an error to the module 10 from which it received the request.)

If the dispatcher 16 locates such an entity node 130 in dispatch table 28, it sequences to the next step (step 191) in the parsing operation, in which attempts to locate an entity node 130 which is associated with instance ⟨node name⟩ as specified in the entity element 183. In that operation, the dispatcher 16 uses the contents of pointer field 152 in the coded entry 131 to locate an entity node with a class/instance flag 140 which identifies the entity node as being associated with instance names and whose coded list includes a coded entry 131 whose instance name entry 132 corresponds to the ⟨node name⟩in entity element 183 of the request 180. Again, if the dispatcher 16 is unable to locate such a node 130 in the dispatch table 28, it searches for a wildcard or ellipsis pointer (see below).

On the other hand, if the dispatcher 16 locates an entity node associated with element 183 in dispatch table 28 in step 191, it sequences to the next step (step 192), in which it attempts to locate an entity node associated with class code 184, ROUTING. In that operation, the dispatcher 16 uses the pointer in field 152 of coded entry 131 and the entity element ROUTING from the request to locate an entity node 130 which includes a class/instance flag 140 which identifies the entity node as being associated with class codes, and whose coded entry list includes a coded entry 131 which has a class code field 151 that contains ROUTING. In that situation, since the entity class ROUTING is an entity class with no instances, the pointer field 152 in the coded entry 131 is null. In this case, the null pointer field 143 in the entity node 130 points to a second entity node 130 associated with the class entity CIRCUIT.

In step 192, the dispatcher 16 uses the null pointer in the entity node 130 associated with the ROUTING class entity located in step 192 to locate a second entity node 130 whose class/instance flag 140 indicates that it is associated with class codes, and a coded list which contains a coded entry 131 whose class code field 151 contains CIRCUIT (step 193). If the dispatcher is unable to locate such an entity node, it searches for a wildcard or ellipsis pointer (see below).

If, on the other hand, the dispatcher 16 locates an entity node 130 in step 193, it sequences to step 194, in which it attempts to locate an entity node 130 identifying the instance entity element ⟨ROUTING CIRCUIT NAME⟩ In that operation, it uses the pointer in field 152 of the coded entry 131 to locate an entity node 130 whose class/instance flag 140 identifies it as being associated with instance names and whose coded list includes a coded entry 132 whose instance name field 151 contains ⟨ROUTING CIRCUIT NAME⟩ as specified in instance entity 186 of the request 180. If the dispatcher 16 is unable to locate such an entry, it searches for a wildcard or ellipsis pointer (see below).

On the other hand, if the dispatcher, in step 194, locates an instance entity node 130 which identifies the instance entity element 186, it has successfully parsed the entity section 182 through 186 of the request 180. Thereafter, the dispatcher 16 uses the pointer in field 152 of the coded entry 131 located in step 194, the verb in verb element 181 and the attributes in characteristics element 187 of the request to identify a dispatch entry 134 (Fig. 8B) to be used in processing the request. In particular, following step 194, the dispatcher 16 uses the pointer in field 152 of coded entry 131 to identify a list of dispatch entries 134. Thereafter, the dispatcher 16 attempts to locate a dispatch entry 134 the contents of whose verb field 165 corresponds to the verb element 181 of the request 180, in this case SHOW, and the contents of whose attribute field 166 corresponds to the attributes in the CHARACTERISTICS element 187.

If the dispatcher 16 locates, in step 195, such a dispatch entry 134, it uses the contents of the procedure identification field 162, process identification field 163, and node identification field 164 to call the procedure to process the request. In this operation, the dispatcher 16 effectively transfers the request to the entity for processing. It will be appreciated that, as described above in connection with Fig. 6, if the process identification in field 163 and node identification in field 164 identify another process or node than contain the dispatcher, the dispatcher transfers the request to the dispatcher in the other process or node, as identified in the respective fields 163 and 164, for processing.

The above describes the use of the coded entries of the dispatch table. The wildcard and ellipsis pointers offer an additional functionality to the table. For example, one management module may handle all requests for modules of a particular global or subordinate entity class. Without wildcard and ellipsis pointers, all of the instances of the class and instances of any subclasses would have to be enumerated in the dispatch table. To avoid this, wildcard and ellipsis pointers are provided, and may be used in a dispatch specification 39A (Fig. 2B) to indicate in a general fashion which entity classes and instances a management module services.

One example of such a dispatch specification is:

NODE * ROUTING CIRCUIT ...

which indicates that the module can handle, for any instance of a NODE class global entity, all instances of the subentity class CIRCUIT as well as all subentities of CIRCUIT class subentities. The asterisk (*) matches any instance name, and the ellipsis (...) matches any instances of the subentity or class/instance pairs of subentities which may follow. For example, the expression

NODE foo ROUTING CIRCUIT bar LINK fred

would match the dispatch specification because "*" would match "foo", and "..." would match "bar LINK fred".

Referring to Fig. 9B, to enter a wildcarded dispatch specification in the dispatch table 28, the entity node 130 at step 191 (Fig. 9A), which corresponds to instance names of NODE class entities, would be modified. The wildcard pointer 141 would be changed to point to a new entity node 130 (step 196) which contained class codes, one of which was the class code ROUTING. The child pointer related to class code ROUTING would be null (as in step 192, Fig. 9A) and the null pointer would point to another new entity node 130 (step 197), which would have a child pointer corresponding to the class name CIRCUIT. This child pointer would point to another new entity node 130 (step 198), whose ellipsis pointer would point to the dispatch entry for the module (step 199).

Parsing of the modified table would be similar to that described by Fig. 9A, until step 191. At step 191, the dispatcher 16 would search for an instance of the NODE class with name, e.g., "foo". If this name was found in the coded entries (three being shown for illustrative purposes) then the dispatcher would proceed according to the child pointers in the coded entries. However, if the name "foo" was not found in the coded entries (indicated by a null NEXT ENTRY pointer in the last coded entry), then the dispatcher would search for a non-null wildcard pointer at step 191. After locating the wildcard pointer, the dispatcher would then proceed to step 196.

Steps 196 and 197 are similar to steps 192 and 193 of Fig. 9A. The dispatcher uses the null pointer in step 196 (corresponding to the class code "ROUTING") to move to step 197, and then uses the child pointer corresponding to the class code "CIRCUIT" to move to step 198.

At step 198, the dispatcher will search the linked list of coded entries (three being shown for illustrative purposes) to locate an instance name of "bar". If this name is not found in the coded entries, the dispatcher then searches for a non-null wildcard pointer. If this is not found, the dispatcher searches for a non-null ellipsis pointer. This will be located, and used to traverse to the dispatch entry (step 199). The contents of the dispatch entry would then be used to call the appropriate module.

Note that the wildcard and ellipsis pointers allow general matching of entity class codes and instance names, but only after the coded entries of the dispatch table are checked. In this way, the dispatcher searches for the "most specific match" of the entity name. Therefore, for example, a first module can have a dispatch specification:

NODE * ROUTING CIRCUIT ...

which indicates that the module can handle, for all instances of a NODE class global entity, all instances of the CIRCUIT class subentity of a ROUTING class subentity. A second module can have a dispatch specification

NODE joe ROUTING CIRCUIT ...

which indicates that the module can handle, for instance "joe" of the NODE class global entity, all instances of the CIRCUIT class subentity of a ROUTING class subentity.

To be consistent with the "most specific match" rule, all directives to NODE joe ROUTING CIRCUIT subentities should be sent to the second module. This is accomplished with the dispatch table schema because the instance name "joe" will appear in the coded entries at step 191, and therefore if "joe" is the instance name in a request to a ROUTING CIRCUIT, the "joe" coded entry will be used (because it is checked first), and the wildcard pointer will not be used.

To properly parse the dispatch tree, a stack must also be used by the dispatcher. A simple example will explain why this is necessary. Consider a new module having the dispatch specification

NODE jim DISKDRIVE ...

which indicates the module can handle, for instance "jim" of the NODE class global entity, all instances of DISKDRIVE class subentities. This specification would be entered in the tree by adding a coded entry at step 191 with the instance name "jim", and adding subsequent new entity nodes, in similar fashion to Fig. 9B. Subsequently, when dispatching requests with global entity class and instance names:

NODE jim

the dispatcher would travel to the new entity nodes.

However, a request with an entity name starting with

NODE jim ROUTING CIRCUIT

could not be serviced by the new module, since the new module only supports DISKDRIVE class subentities for NODE instance "jim". Therefore, once the dispatcher determines that the class name ROUTING CIRCUIT is not supported by the new module, it must have a mechanism for returning to step 191, and potentially using other coded entries or the wildcard or ellipsis pointers to find a module which will service the "NODE jim ROUTING CIRCUIT" request. Therefore, as the dispatcher traverses the dispatch table, it maintains a stack of pointers to all of the entity nodes 130 which it has traversed from the root node. Pointers are pushed onto and popped off of this stack as the dispatcher moves up and down through the dispatch table tree structure attempting to find the appropriate dispatch entry.

If no matching dispatch entry is found, an error is returned to the requestor (i.e. presentation or functional module).

As discussed above, a control functional module may serve as a pass-through from the presentation modules directly to the access modules. To implement such a pass-through, the ellipsis pointer for the root node of the presentation-function aspect of the dispatch table (which will match any entity name in any request) should point to the dispatch entry for the control functional module. Whenever it receives a request, the control functional module will simply issue an identical request to the function-access aspect of the dispatcher. In this way, all requests which do not match dispatch specifications in the presentation-function dispatch table will also be routed for matching in the function-access dispatch table. This allows presentation module requests to access primitive functions available from the access modules.

In an alternative embodiment of the dispatch table, to allow more than one class code which doesn't have instances, the null pointer field 143 may contain the first element of a linked list similar in structure to the list of coded entries 131. The second, "null" list would contain code values of class codes which have no instances. The null list would be parsed after the coded list, but before checking for a wildcard pointer.

G. DOMAINS AND CONFIGURATION

As described above, a configuration functional module 11 maintains a configuration database defining the entities comprising the complex system. By means of appropriate commands from an operator, the configuration functional module 11 can add instances of entities, as defined in the data dictionary, to the configuration database, delete them from the configuration database, and change the definitions in the configuration database. As also described above, a domain functional module 11 establishes a domain entity in the configuration database which refers to a subset of the entities already defined in the configuration database. An operator, through a presentation module 10, can control and monitor the entities comprising a specific domain, without regard to the possibly myriad other entities comprising the complex system. In addition, an operator can initiate a control or monitoring operation in connection with entities only in the domain, without having to initiate generation of a request by a presentation module 10 for each entity, thereby simplifying control and monitoring of the complex system.

The domain functional module 11 establishes, within or in addition to the configuration database, a domain database for each domain entity, identifying the entities comprising the domain entity. Upon receipt of an appropriate request, the domain functional module 11 will add an entity to a domain database, thereby adding the entity to the domain, delete an entity from a domain database, thereby deleting the entity from the domain, generate a response identifying the entities comprising a domain as identified in the domain database, and delete a domain database, thereby effectively deleting the domain.

Referring to Fig. 9C, the format of the configuration and domain databases (which may be incorporated in a single database) includes a field for each entity instance in the configuration, and similarly, each entity instance in the domain.

The domain database includes an entry 230 for each member of a domain, listing the domain name and the instance name of the entity or subentity member. In addition, the domain database includes an entry 232 for each entity which is a member of any domain, listing the instance name and the domains which it is a member. The domain functional module updates this information as the domains are modified, and can use the information to quickly determine the members of a domain, or, alternatively, to quickly determine the domain membership of an entity.

In alternative embodiments, a first domain may incorporate the members of a second domain by reference to the second domain, thus reducing the size of the domains database. In other embodiments, the domains database may establish a hierarchy of domains similar to the hierarchy of entities and subentities, and commands may be directed similarly to domains and subdomains.

The configuration database includes an entry 234 for each entity and subentity, organized hierarchically in the database. The full name for each entity and subentity instance is provided. This information can be used by the configuration functional module to quickly determine the configuration, for example, to display (via a presentation module) to the user a map of the configuration or menus of entity instance names.

H. ALARMS

As described above in connection with Fig. 1B, one functional module 11 comprises an alarms functional module 11, which can establish alarm conditions, in response to requests from a presentation module 10, and, using the various conditions of the entities of the complex system, as, for example, recorded in the user interface information file 29, detect the occurrence of an alarm condition.

Fig. 10A depicts the functional organization of the alarms functional module 11. With reference to Fig. 10A, the alarms functional module 11 includes a general alarms module 200 that receives requests from the module, interprets them and enables one or more detector modules 201 or one or more rule maintenance modules 202 to operate in response thereto. As indicated above, the alarms functional module 11 performs two general types of operations, namely, maintenance of alarm conditions and detection of alarm conditions.

The maintenance of alarm conditions operation of the alarm functional module 11 is performed by the rule maintenance module 202, which maintains, in an alarm rule base 203, rules which identify each of the alarm conditions. Each rule represents the set of conditions which must be evaluated to determine the existence of an alarm condition. Specifically, the rule maintenance module 202 generates, in response to requests from a presentation module 10, rules, as described below in connection with Fig. 10B, which are stored in the alarm rule base 203. In addition, the rule maintenance module 202, in response to corresponding requests from a presentation module 10, may modify the rules in the alarm rule base 203, which thereby results in modification of the conditions under which an alarm condition, as represented by the rule, will exist.

Similarly, the operation of detection of alarm conditions is performed by the detector module 201, which uses e.g., the condition information in the historical data file (Fig. 5) and the alarm rules in the alarm rule base 203. As described below in connection with Fig. 10B, each rule includes a condition portion, which identifies the conditions. The detector module 201, to detect an alarm condition, determines whether e.g., the contents of the historical data file match the conditions of the various rules. If so, the detector module 201 generates an alarm indication, for transfer by the general alarms module 200 via a notification module 204 to, e.g., presentation module 10 for display to the operator.

The general form of an alarm rule, as generated by the rule maintenance module 202, is depicted in Fig. 10B. With reference to Fig. 10B, an alarm rule includes a condition portion 210, which sets forth the set of condition(s) required for the indication of the alarm. The condition portion includes an expression portion 212, a relational operator 213 and an expression value portion 214. The relational operator 213 relates the expression portion 212 to the expression value portion 214, so that the condition portion 210 evaluates to either a logical TRUE or a logical FALSE. It will be appreciated that, if the expression portion 212 itself evaluates to logical TRUE or logical FALSE, the relational operator 213 and expression value portion 214 of the condition portion 210 are not needed. In either case, if the condition portion evaluates to a logical TRUE, an alarm condition exists.

The rule includes an entity and attribute portion 212 and a time value portion 216. The rel-op value portion 213 relates values of one attribute to one value portion 214. The time value portion 216 establishes a time function, and may indicate the times or time intervals at which the condition portion 210 is to be used by the alarm detector module 201.

Providing an alarms functional module 11 permits an operator to establish alarm conditions on a dynamic or as-needed basis. Since the alarm conditions do not have to be pre-established in the control arrangement, the control arrangement can be used in controlling and monitoring a wide variety of diverse complex systems. For example, if the control arrangement is being used to control and monitor distributed digital data processing systems, which may have diverse configurations of nodes communicating over a network, the alarm conditions can be established by an operator based on the particular configuration. In addition, alarm conditions can be added by addition of rules to the alarm rule base 203, if a new alarm condition is discovered during operation of the complex system.

I. OTHER EMBODIMENTS

The foregoing description has been limited to a specific embodiment of this invention. It will be apparent, however, that variations and modifications may be made to the invention, with the attainment of some or all of the advantages of the invention. Therefore, it is the object of the appended claims to cover all such variations and modifications as come within the true spirit and scope of the invention.

Claims

1. A system for controlling and carrying cut management functions over an assemblage of entities, wherein said entities interface within said assemblage for control of primary information handling functions and said entities further interface with said system to permit the carrying out of said management functions, said management functions including a function for retrieval of one or more attribute values from an entity, said attribute values identifying the state of operation of entity, said system comprising:

   stored management modules carrying out said management functions by executing selected management-related commands,
   storage containing records of said attribute values, each record including an indicator of an associated time; and
   at least one said module storing rules identifying attribute values at one or more times corresponding to alarm conditions and comprising a rule generator for generating rules for storage and an alarm condition detector for retrieving attribute values from storage or said entities and detecting an alarm condition in response to the contents of said rules if retrieved attribute values correspond to an alarm condition.

2. The system of claim 1, wherein:
   said management modules carry out said management functions by independently interpreting and executing selected management-related commands.

3. The system of claim 1, wherein:
   at least some said management functions generate, for display to a user, management information indicating the status of said primary information handling functions of one or more said entities.

4. The system of claim 1 further comprising:

   a historical data recorder for periodically accessing and storing new attribute values in said records in response to a predetermined schedule.

5. The system of claim 1, wherein said rule module is responsive to user commands for generating new rules or modifying existing rules.

6. The system of claim 1, wherein said rules may be enabled or disabled in response to user commands.

7. The system of claim 1, wherein the syntax of said rules comprises an entity and attribute specifier, a relational operator, a value, and a time parameter.

8. A method for controlling and carrying out management functions over an assemblage of entities, wherein said entities interface within said assemblage for control of primary information handling functions and said entities further interface with said system to permit the carrying out of said management functions, said management functions including a function for retrieval of one or more attribute values from an entity, said attribute value identifying the state of operation of said entity, comprising:

   carrying out said management functions by executing selected management-related commands within management modules;
   storing records of said attribute values, each record including an indication of an associated time; and
   in at least one said module:

      storing rules identifying attribute values at one or more times corresponding to alarm conditions,
      generating rules for storage; and
      retrieving attribute values from storage or said entities and detecting an alarm condition in response to the

contents of said rules if retrieved values correspond to an alarm condition.

9. The method of claim 8 further comprising:

a historical data recorder operating in one member of said computer network for periodically accessing and storing new attribute values into records at said one member in response to a predetermined schedule.

10. The method of claim 8, wherein:
at least some said management functions generate, for display to a user, management information indicating the status of said primary information handling functions of one or more said members.

11. The method of claim 8, wherein said module storing rules is responsive to user commands for generating new rules ore modifying existing ones.

12. The method of claim 8, wherein a syntax of said rules comprises a member and attribute specifier, a relational operator, a value, and a time parameter.

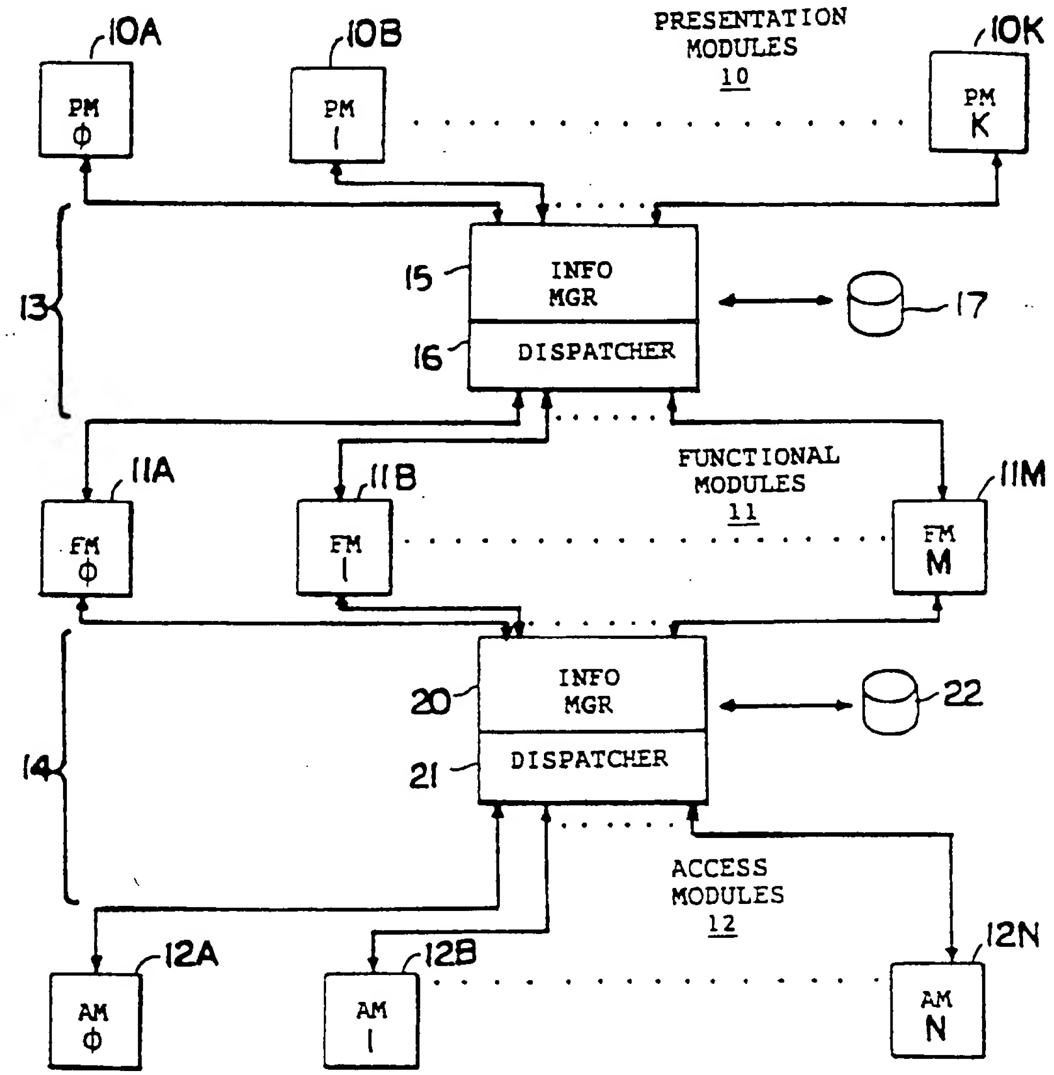

10A
10B
PRESENTATION MODULES 10
10K
PM Φ
PM 1
PM K
13
15
INFO MGR
16
DISPATCHER
17
11A
11B
FUNCTIONAL MODULES 11
11M
FM Φ
FM 1
FM M
14
20
INFO MGR
21
DISPATCHER
22
ACCESS MODULES 12
12A
12B
12N
AM Φ
AM 1
AM N

FIG. 1A

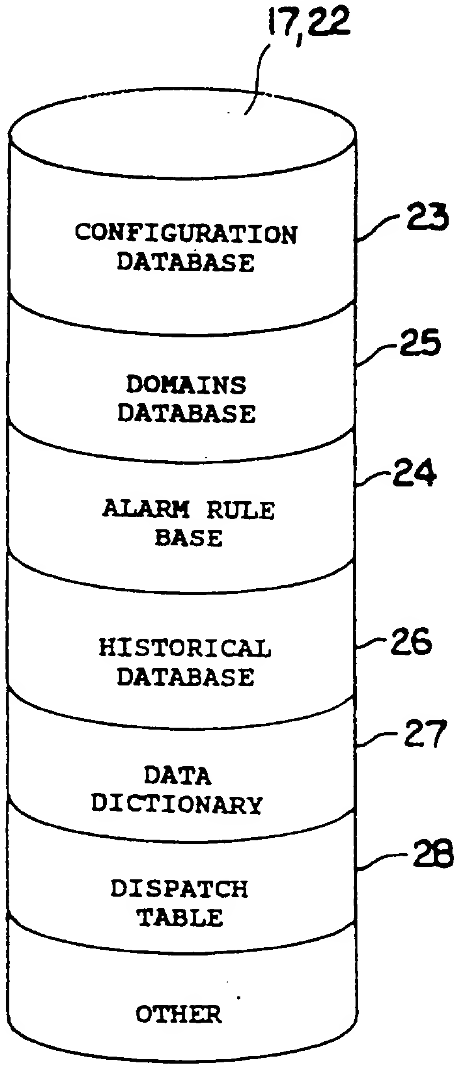

17,22
CONFIGURATION DATABASE
23
DOMAINS DATABASE
25
ALARM RULE BASE
24
HISTORICAL DATABASE
26
DATA DICTIONARY
27
DISPATCH TABLE
28
OTHER

FIG. 1B

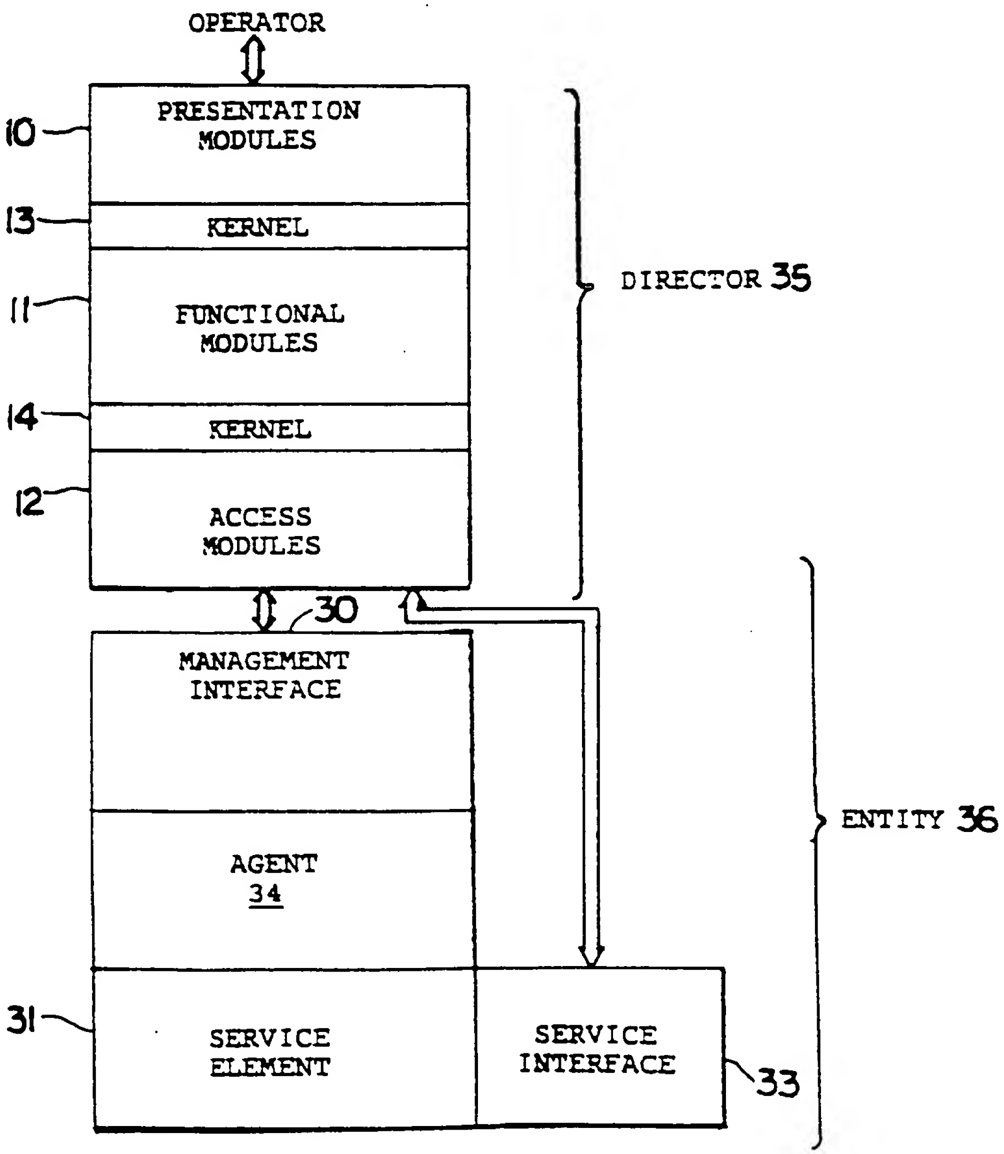

OPERATOR
10
PRESENTATION MODULES
13
KERNEL
11
FUNCTIONAL MODULES
14
KERNEL
12
ACCESS MODULES
DIRECTOR 35
30
MANAGEMENT INTERFACE
AGENT 34
ENTITY 36
31
SERVICE ELEMENT
SERVICE INTERFACE
33

FIG. 2A

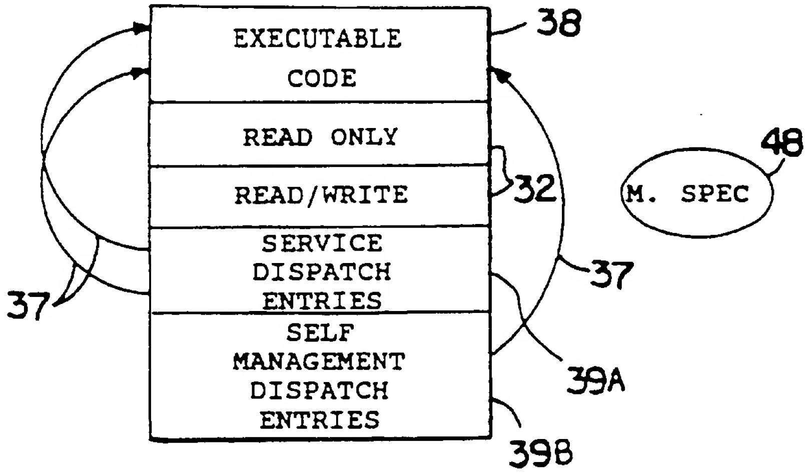
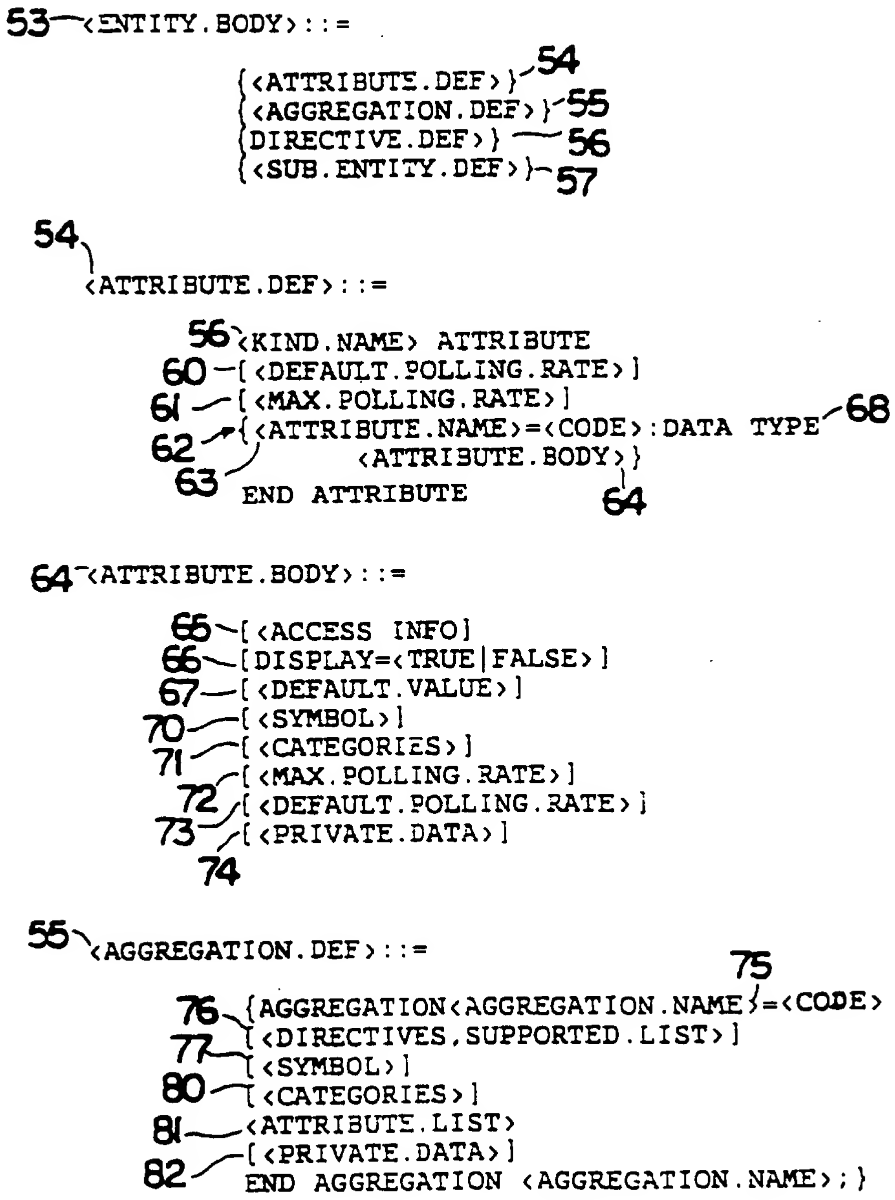

EXECUTABLE CODE
38
READ ONLY
READ/WRITE
32
48
M. SPEC
SERVICE DISPATCH ENTRIES
37
37
SELF MANAGEMENT DISPATCH ENTRIES
39A
39B

FIG. 2B

```
    <MANAGEMENT SPECIFICATION">::=                        41
48              MANAGEMENT SPECIFICATION <SPEC. NAME>;
                      <VERSION>—42
           40         <FACILITY>—43
                      <TYPE.DECLARATION>—44
                      <SPEC.BODY>—45
                END SPECIFICATION [<SPEC.NAME>]

          45 <SPEC.BODY>::=
                      {<GLOBAL.ENTITY.DEF>}—45A

                      {<SUBORDINATE.ENTITY.DEF>}—45C

   46
    <ENTITY.DEF>::=<GLOBAL|SUBORDINATE>
                                                              47
                            ENTITY<CLASS.NAME>=<CODE>:
          50—[SUPERIOR = <CLASS.NAME>[<CLASS.NAME>;]
          51—IDENTIFIER = (<ATTRIBUTE.LIST>),
          52—[SYMBOL = <STRING>:]
          53—<ENTITY.BODY>
               END ENTITY <CLASS.NAME>;
```

FIG. 3A



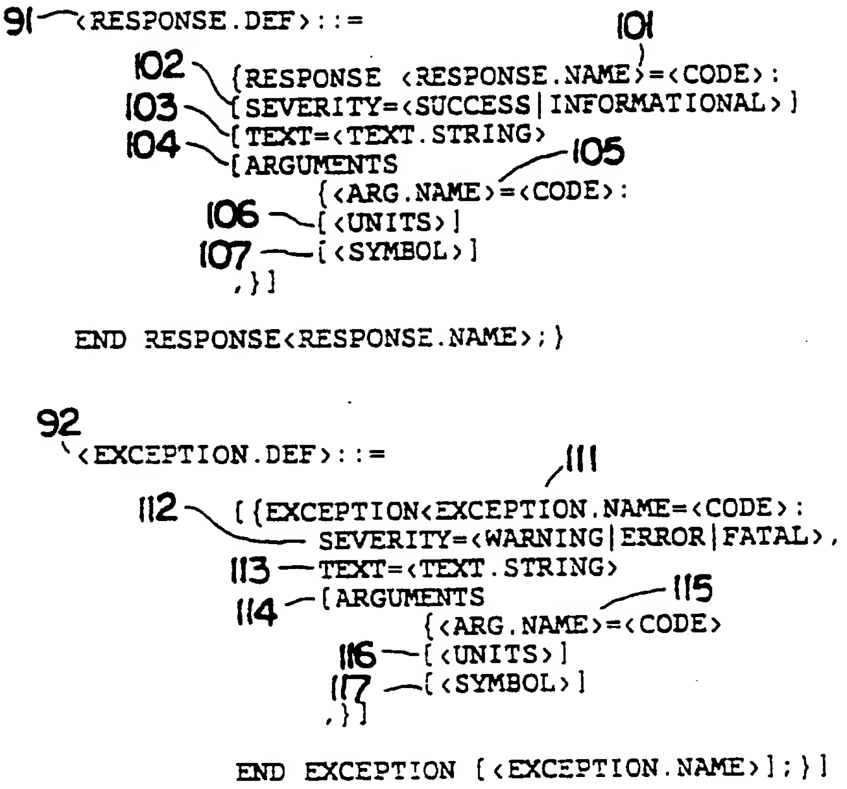
```
<ENTITY.BODY>::=
                    {<ATTRIBUTE.DEF>}
                    {<AGGREGATION.DEF>}
                    {DIRECTIVE.DEF>}
                    {<SUB.ENTITY.DEF>}

<ATTRIBUTE.DEF>::=
          <KIND.NAME> ATTRIBUTE
          [<DEFAULT.POLLING.RATE>]
          [<MAX.POLLING.RATE>]
          {<ATTRIBUTE.NAME>=<CODE>:DATA TYPE
                 <ATTRIBUTE.BODY>}
          END ATTRIBUTE

<ATTRIBUTE.BODY>::=
          [<ACCESS INFO]
          [DISPLAY=<TRUE|FALSE>]
          [<DEFAULT.VALUE>]
          [<SYMBOL>]
          [<CATEGORIES>]
          [<MAX.POLLING.RATE>]
          [<DEFAULT.POLLING.RATE>]
          [<PRIVATE.DATA>]

<AGGREGATION.DEF>::=
          {AGGREGATION<AGGREGATION.NAME>=<CODE>
          [<DIRECTIVES,SUPPORTED.LIST>]
          [<SYMBOL>]
          [<CATEGORIES>]
          <ATTRIBUTE.LIST>
          [<PRIVATE.DATA>]
          END AGGREGATION <AGGREGATION.NAME>;}
```

FIG. 3B

```
56 <DIRECTIVE.DEF>::=                                  83
            {DIRECTIVE<DIRECTIVE.NAME>=<CODE>
        84 [ACTION-DIRECTIVE=<TRUE|FALSE>,]
        85 [DISPLAY=<TRUE|FALSE>,]
        86 [<SYMBOL>]
        87 [<CATEGORIES>]
        90 <REQUEST.DEF>
        91 <RESPONSE.DEF>
        92 <EXCEPTION.DEF>
            END DIRECTIVE <DIRECTIVE.NAME>;}

90 <REQUEST.DEF>::=
            REQUEST
            91 [ARGUMENTS        92
                    {<ARG.NAME>=<CODE>
               93 [DISPLAY=<TRUE|FALSE>]
               94 [<REQUIRED>]
               95 [<UNITS>]
               96 [<DEFAULT>]
               97 [<SYMBOL>]
              100 [<PRIVATE.DATA>]
                  j }]
            END REQUEST;
```

FIG. 3C

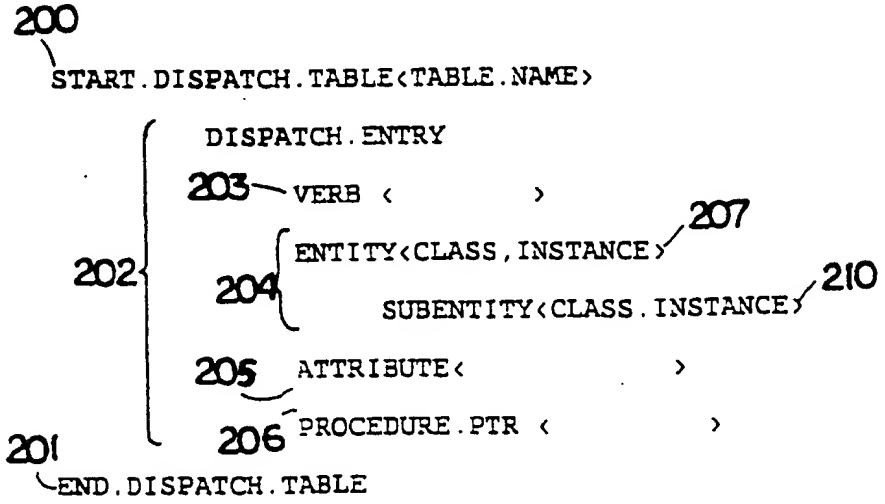
FIG. 3D



```
<RESPONSE.DEF>::=

        {RESPONSE <RESPONSE.NAME>=<CODE>:
        [SEVERITY=<SUCCESS|INFORMATIONAL>]
        [TEXT=<TEXT.STRING>
        [ARGUMENTS
              {<ARG.NAME>=<CODE>:
              [<UNITS>]
              [<SYMBOL>]
          ,}]

END RESPONSE<RESPONSE.NAME>;}

<EXCEPTION.DEF>::=

        [{EXCEPTION<EXCEPTION.NAME=<CODE>:
              SEVERITY=<WARNING|ERROR|FATAL>,
              TEXT=<TEXT.STRING>
              [ARGUMENTS
                     {<ARG.NAME>=<CODE>
                     [<UNITS>]
                     [<SYMBOL>]
              ,}]

        END EXCEPTION [<EXCEPTION.NAME>];}]
```

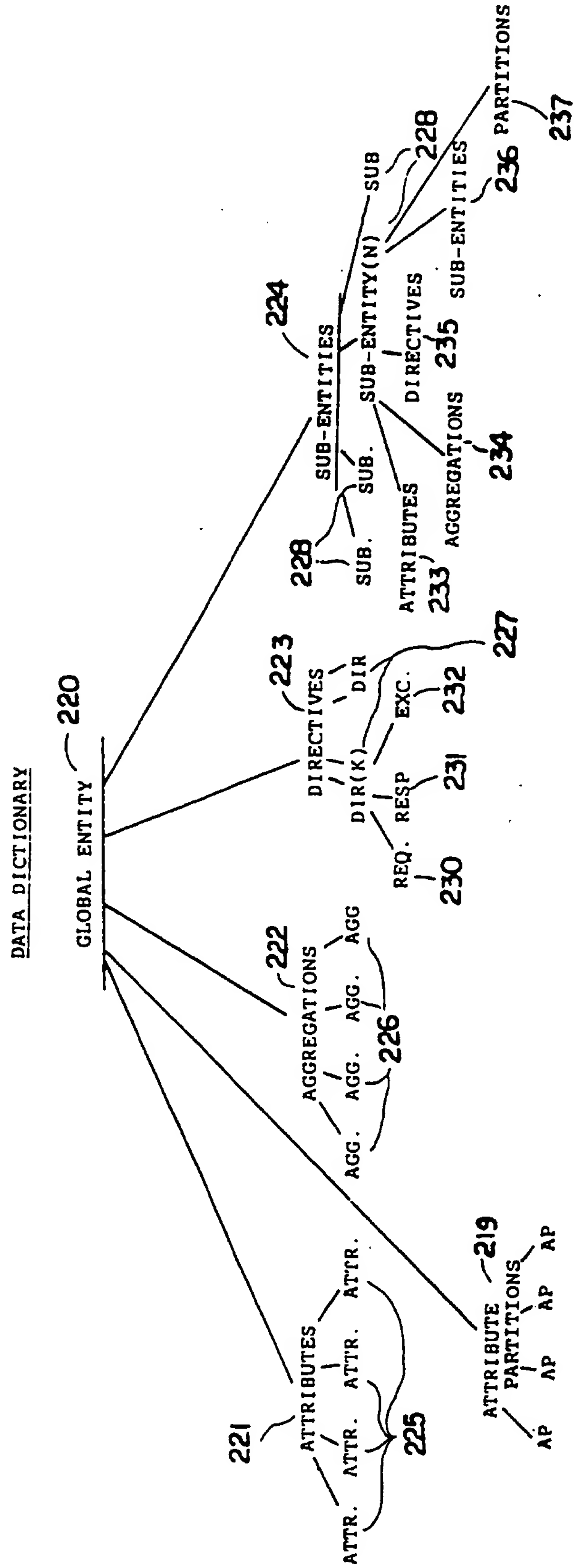
FIG. 3E

DISPATCH SPECIFICATION



```
START.DISPATCH.TABLE<TABLE.NAME>

        DISPATCH.ENTRY
        VERB <            >
        ENTITY<CLASS,INSTANCE>
              SUBENTITY<CLASS.INSTANCE>
        ATTRIBUTE<            >
        PROCEDURE.PTR <            >
END.DISPATCH.TABLE
```

FIG. 4


DATA DICTIONARY
GLOBAL ENTITY
220
ATTRIBUTES
221
ATTR.
ATTR.
ATTR.
ATTR.
225
ATTRIBUTE PARTITIONS
219
AP
AP
AP
AP
AGGREGATIONS
222
AGG.
AGG.
AGG.
AGG
226
DIRECTIVES
223
DIR(K)
DIR
227
REQ.
230
RESP
231
EXC.
232
SUB-ENTITIES
224
SUB.
SUB.
SUB-ENTITY(N)
SUB
228
ATTRIBUTES
233
AGGREGATIONS
234
DIRECTIVES
235
SUB-ENTITIES
236
PARTITIONS
237

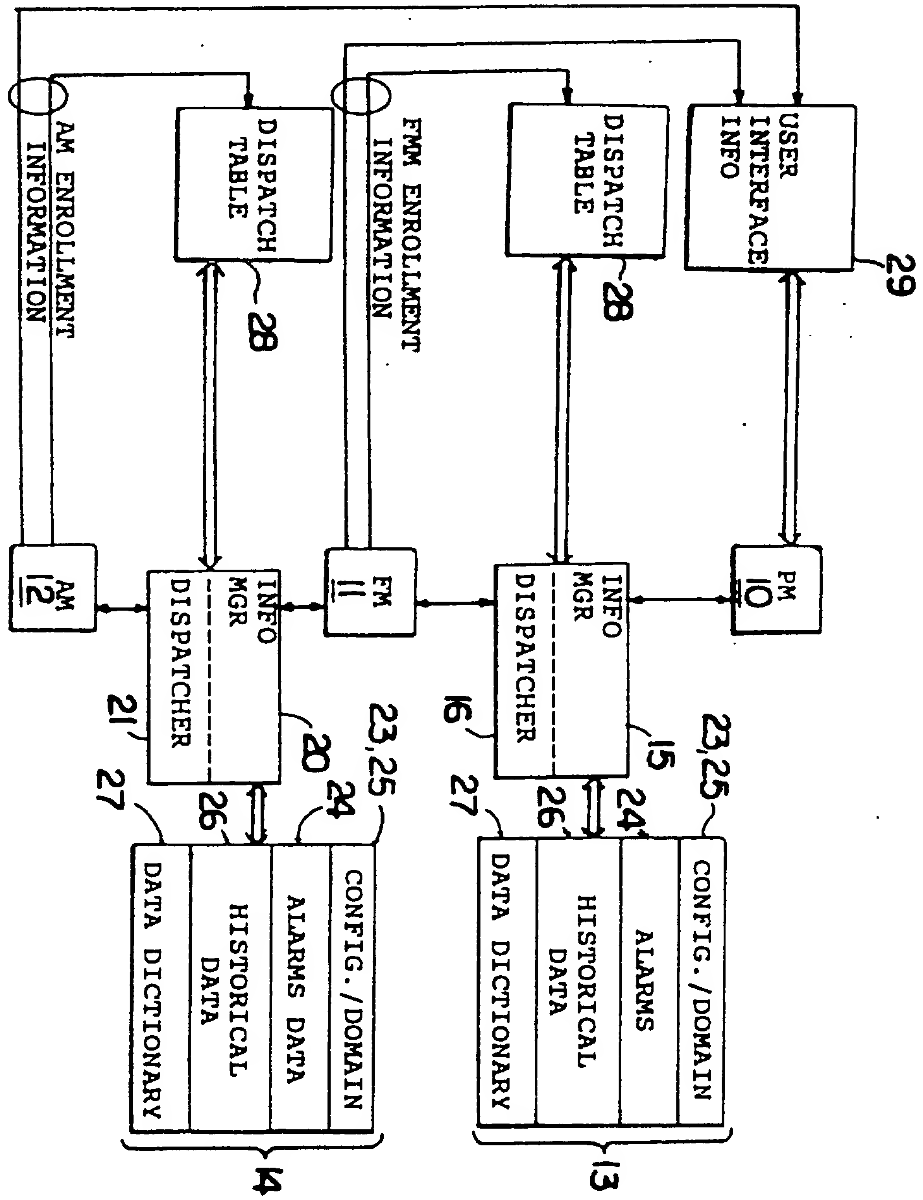

USER INTERFACE INFO
29
DISPATCH TABLE
28
DISPATCH TABLE
28
FMM ENROLLMENT INFORMATION
AM ENROLLMENT INFORMATION
PM 10
INFO MGR
DISPATCHER
15
16
FM 11
INFO MGR
DISPATCHER
20
21
AM 12
23,25
24
26
27
CONFIG./DOMAIN
ALARMS
HISTORICAL DATA
DATA DICTIONARY
13
23,25
24
26
27
CONFIG./DOMAIN
ALARMS DATA
HISTORICAL DATA
DATA DICTIONARY
14

FIG. 5

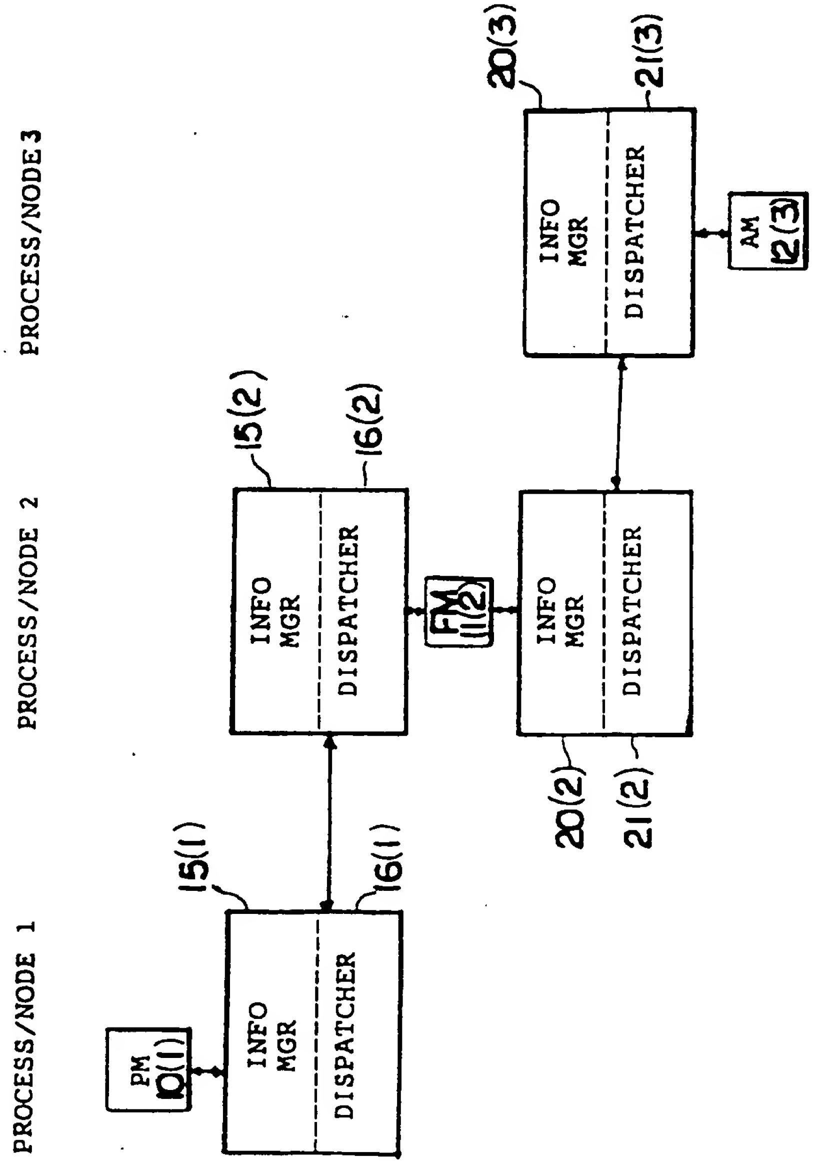

PROCESS/NODE 1
PROCESS/NODE 2
PROCESS/NODE 3
PM
10(1)
INFO MGR
DISPATCHER
15(1)
16(1)
INFO MGR
DISPATCHER
15(2)
16(2)
FM
11(2)
INFO MGR
DISPATCHER
20(2)
21(2)
INFO MGR
DISPATCHER
20(3)
21(3)
AM
12(3)

FIG. 6

REQUEST/SUBSIDIARY
REQUEST PARAMETER:

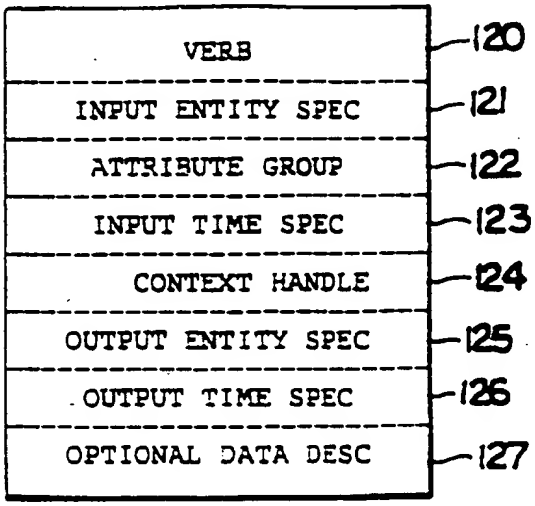


FIG. 7A

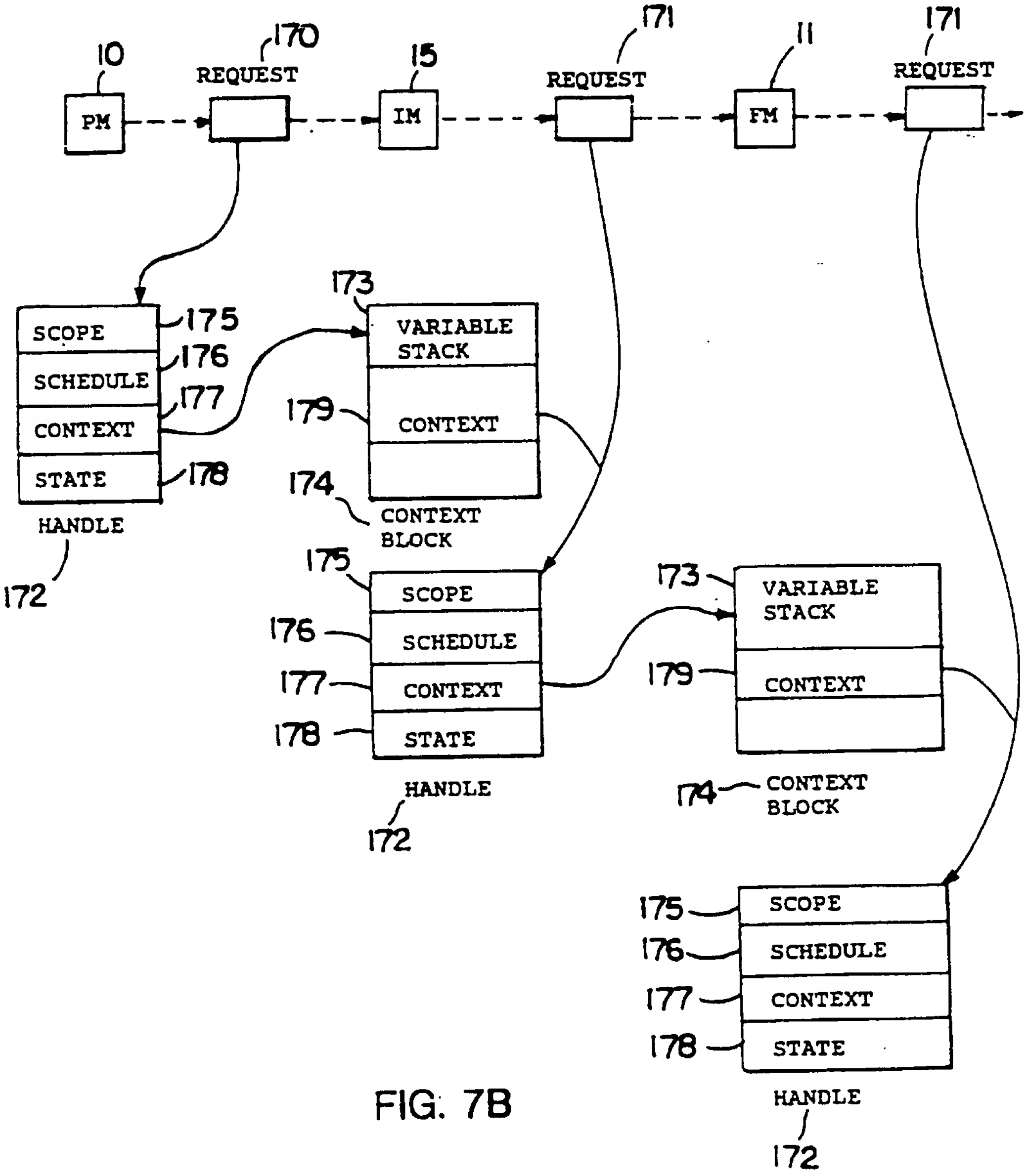

10
PM
170
REQUEST
15
IM
171
REQUEST
11
FM
171
REQUEST
SCOPE
SCHEDULE
CONTEXT
STATE
HANDLE
172
175
176
177
178
173
VARIABLE STACK
179
CONTEXT
174
CONTEXT BLOCK
175
SCOPE
176
SCHEDULE
177
CONTEXT
178
STATE
HANDLE
172
173
VARIABLE STACK
179
CONTEXT
174
CONTEXT BLOCK
175
SCOPE
176
SCHEDULE
177
CONTEXT
178
STATE
HANDLE
172

FIG. 7B

FIG. 8A

DISPATCH TREE
ENTITY NODE 130

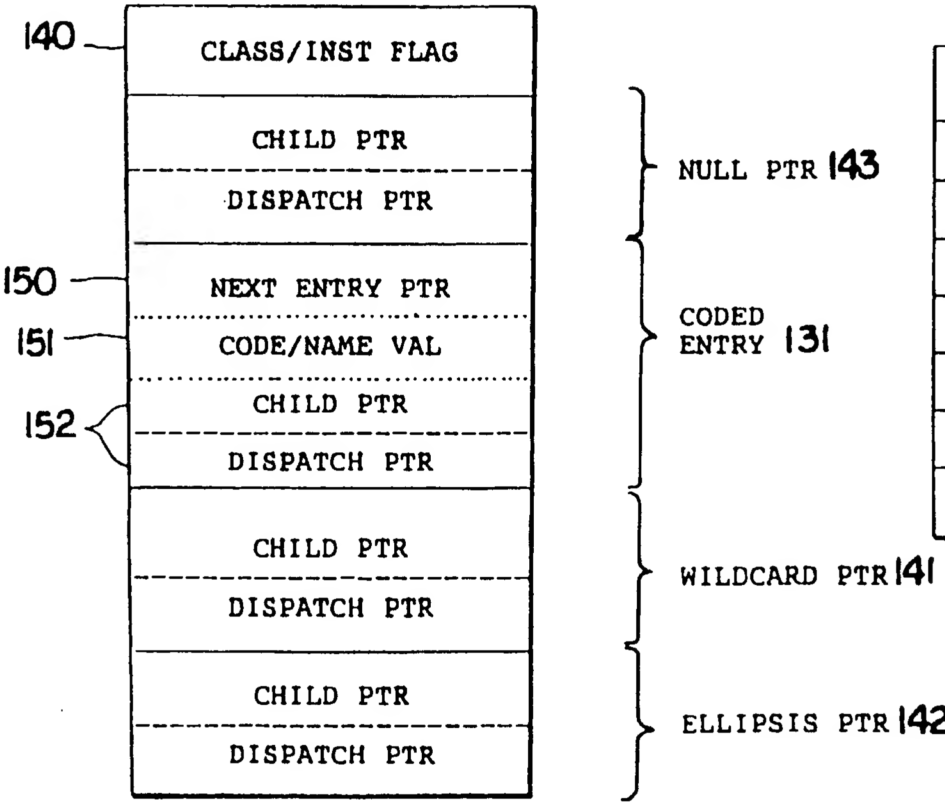
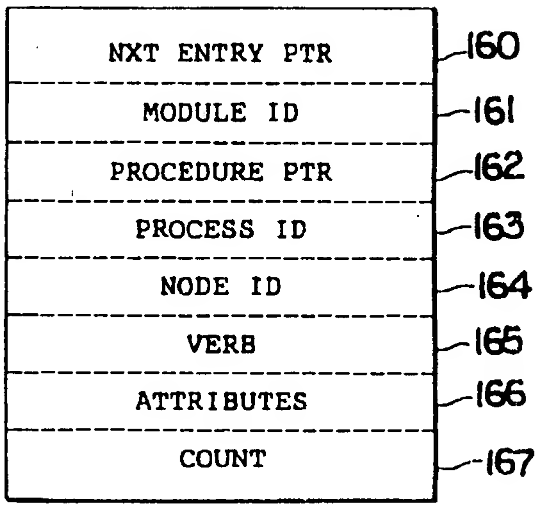


FIG. 8B

DISPATCH ENTRY 134



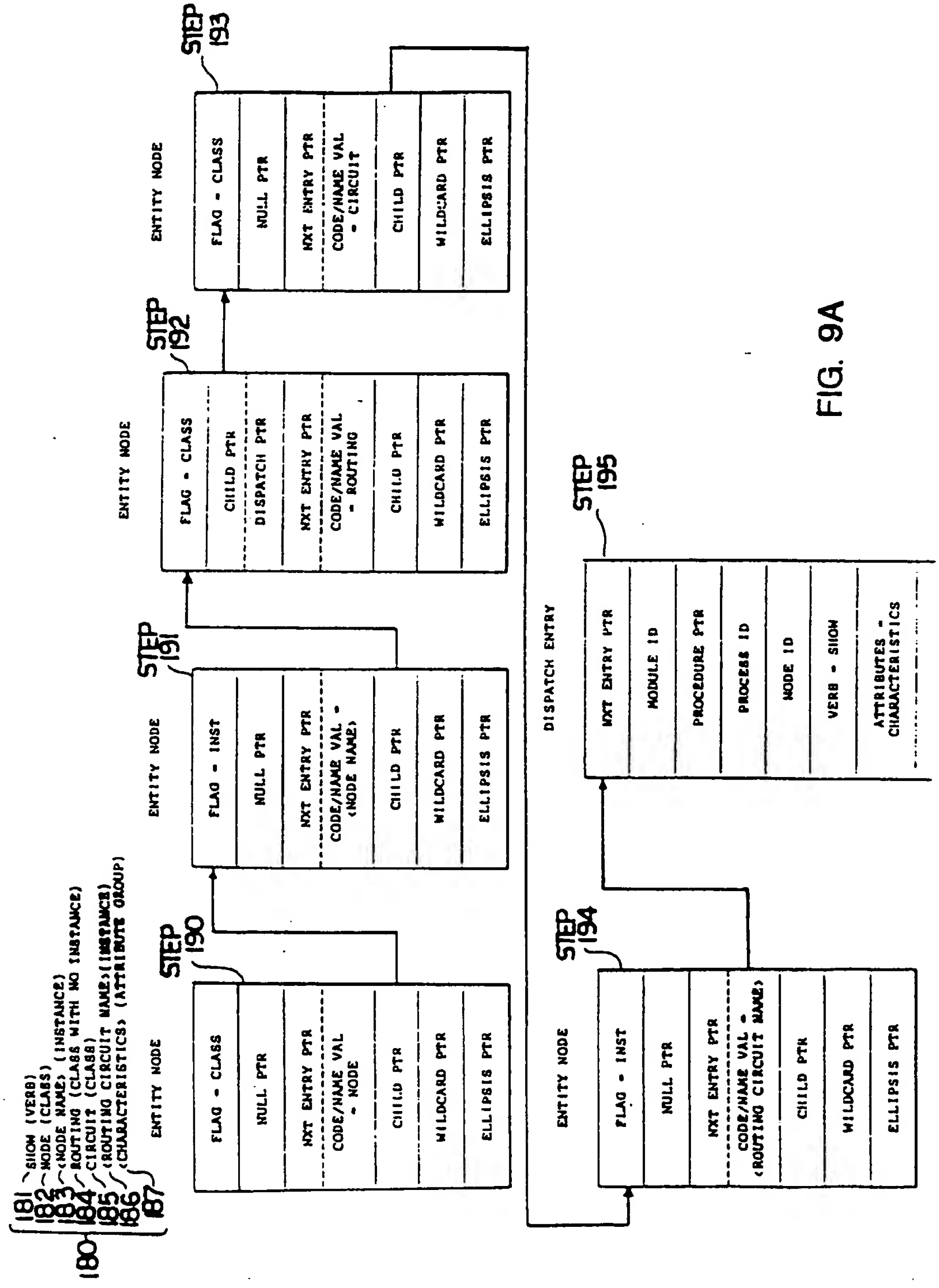

180
181 SHOW (VERB)
182 NODE (CLASS)
183 <NODE NAME> (INSTANCE)
184 ROUTING (CLASS WITH NO INSTANCE)
185 CIRCUIT (CLASS)
186 <ROUTING CIRCUIT NAME>(INSTANCE)
187 <CHARACTERISTICS> (ATTRIBUTE GROUP)
STEP 190
ENTITY NODE
FLAG - CLASS
NULL PTR
NXT ENTRY PTR
CODE/NAME VAL - NODE
CHILD PTR
WILDCARD PTR
ELLIPSIS PTR
STEP 191
ENTITY NODE
FLAG - INST
NULL PTR
NXT ENTRY PTR
CODE/NAME VAL - <NODE NAME>
CHILD PTR
WILDCARD PTR
ELLIPSIS PTR
STEP 192
ENTITY NODE
FLAG - CLASS
CHILD PTR
DISPATCH PTR
NXT ENTRY PTR
CODE/NAME VAL - ROUTING
CHILD PTR
WILDCARD PTR
ELLIPSIS PTR
STEP 193
ENTITY NODE
FLAG - CLASS
NULL PTR
NXT ENTRY PTR
CODE/NAME VAL - CIRCUIT
CHILD PTR
WILDCARD PTR
ELLIPSIS PTR
STEP 194
ENTITY NODE
FLAG - INST
NULL PTR
NXT ENTRY PTR
CODE/NAME VAL - <ROUTING CIRCUIT NAME>
CHILD PTR
WILDCARD PTR
ELLIPSIS PTR
STEP 195
DISPATCH ENTRY
NXT ENTRY PTR
MODULE ID
PROCEDURE PTR
PROCESS ID
NODE ID
VERB - SHOW
ATTRIBUTES - CHARACTERISTICS

FIG. 9A

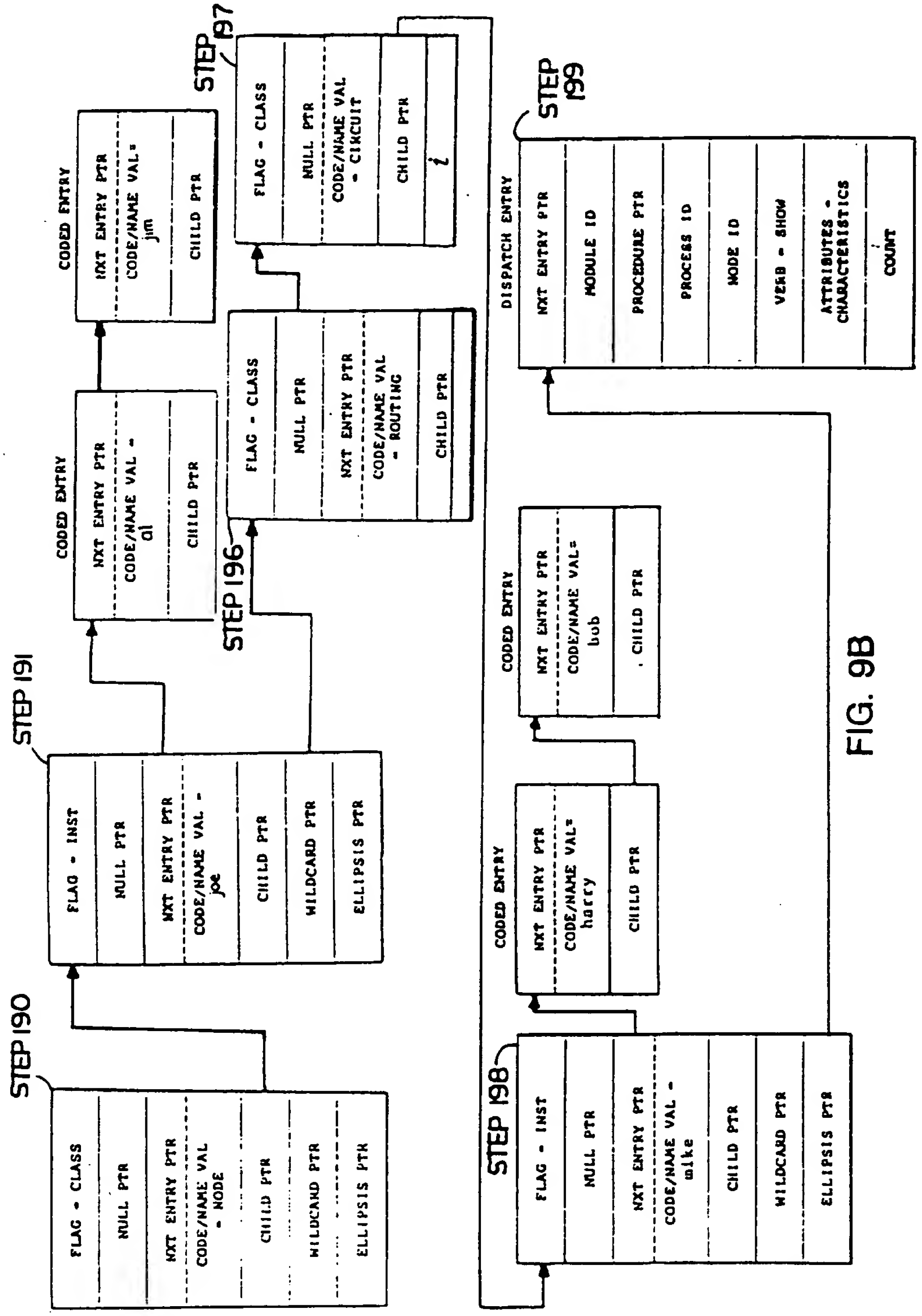
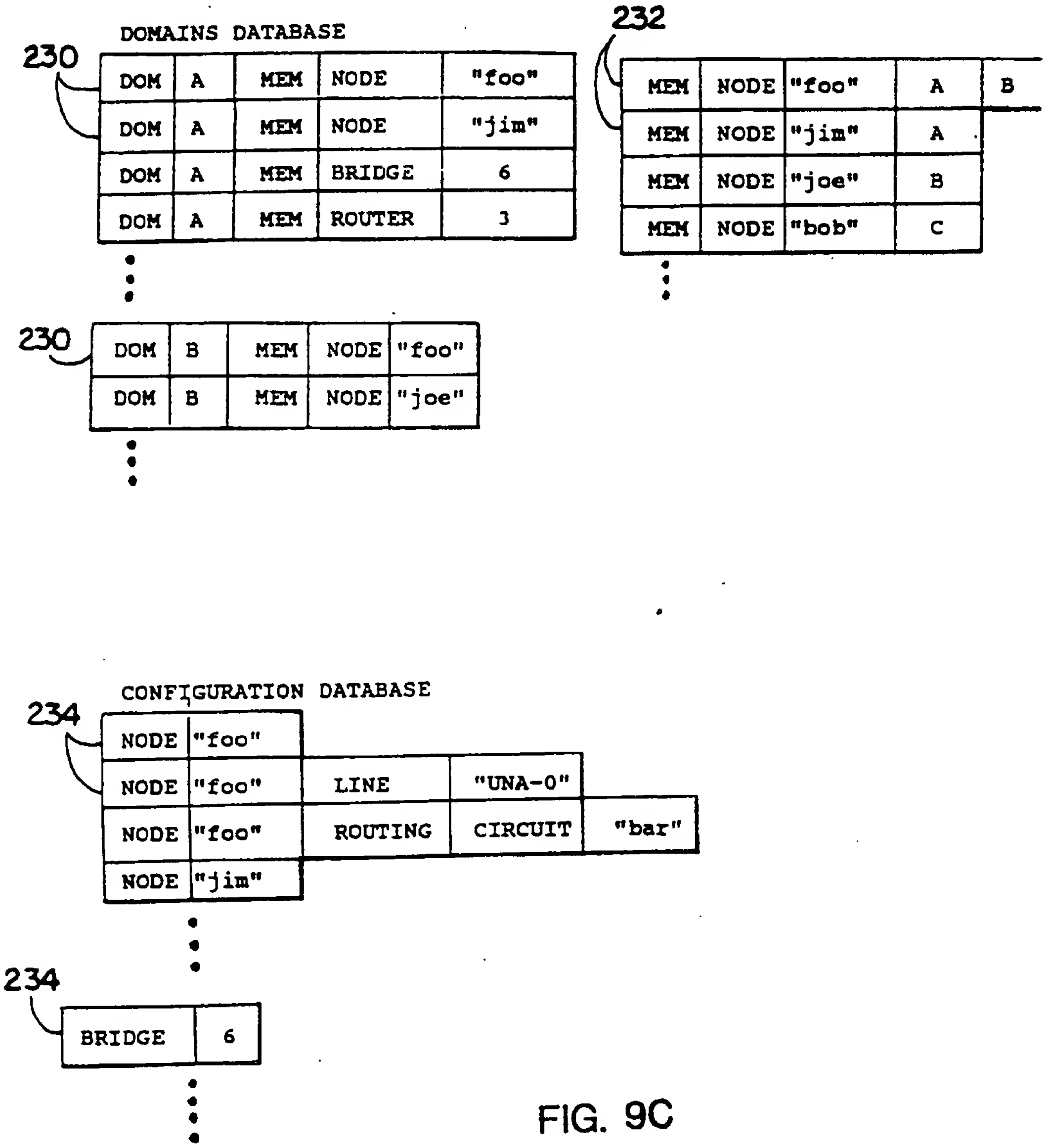

STEP 190
FLAG - CLASS
NULL PTR
NXT ENTRY PTR
CODE/NAME VAL - NODE
CHILD PTR
WILDCARD PTR
ELLIPSIS PTR
STEP 191
FLAG - INST
NULL PTR
NXT ENTRY PTR
CODE/NAME VAL - joe
CHILD PTR
WILDCARD PTR
ELLIPSIS PTR
CODED ENTRY
NXT ENTRY PTR
CODE/NAME VAL - al
CHILD PTR
CODED ENTRY
NXT ENTRY PTR
CODE/NAME VAL= jim
CHILD PTR
STEP 196
FLAG - CLASS
NULL PTR
NXT ENTRY PTR
CODE/NAME VAL - ROUTING
CHILD PTR
STEP 197
FLAG - CLASS
NULL PTR
CODE/NAME VAL - CIRCUIT
CHILD PTR
STEP 198
FLAG - INST
NULL PTR
NXT ENTRY PTR
CODE/NAME VAL - mike
CHILD PTR
WILDCARD PTR
ELLIPSIS PTR
CODED ENTRY
NXT ENTRY PTR
CODE/NAME VAL= harry
CHILD PTR
CODED ENTRY
NXT ENTRY PTR
CODE/NAME VAL= bob
CHILD PTR
STEP 199
DISPATCH ENTRY
NXT ENTRY PTR
MODULE ID
PROCEDURE PTR
PROCESS ID
NODE ID
VERB - SHOW
ATTRIBUTES - CHARACTERISTICS
COUNT

FIG. 9B



DOMAINS DATABASE

DOM	A	MEM	NODE	"foo"
DOM	A	MEM	NODE	"jim"
DOM	A	MEM	BRIDGE	6
DOM	A	MEM	ROUTER	3

DOM	B	MEM	NODE	"foo"
DOM	B	MEM	NODE	"joe"

MEM	NODE	"foo"	A	B
MEM	NODE	"jim"	A	
MEM	NODE	"joe"	B	
MEM	NODE	"bob"	C	

CONFIGURATION DATABASE

NODE	"foo"			
NODE	"foo"	LINE	"UNA-0"	
NODE	"foo"	ROUTING	CIRCUIT	"bar"
NODE	"jim"			

BRIDGE	6

FIG. 9C

FIG. 10A

ALARMS FUNCTIONAL MODULE

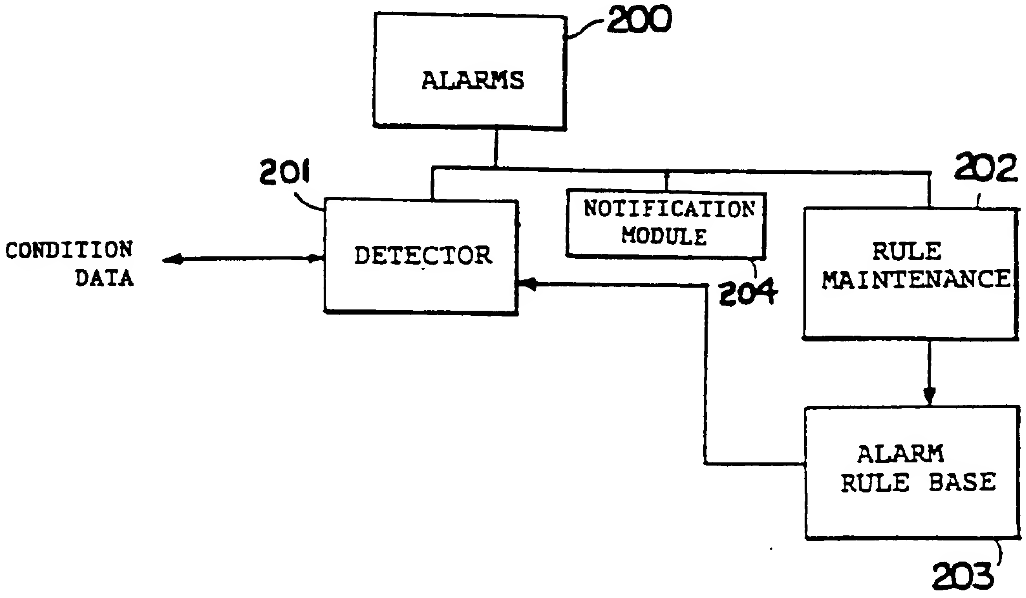
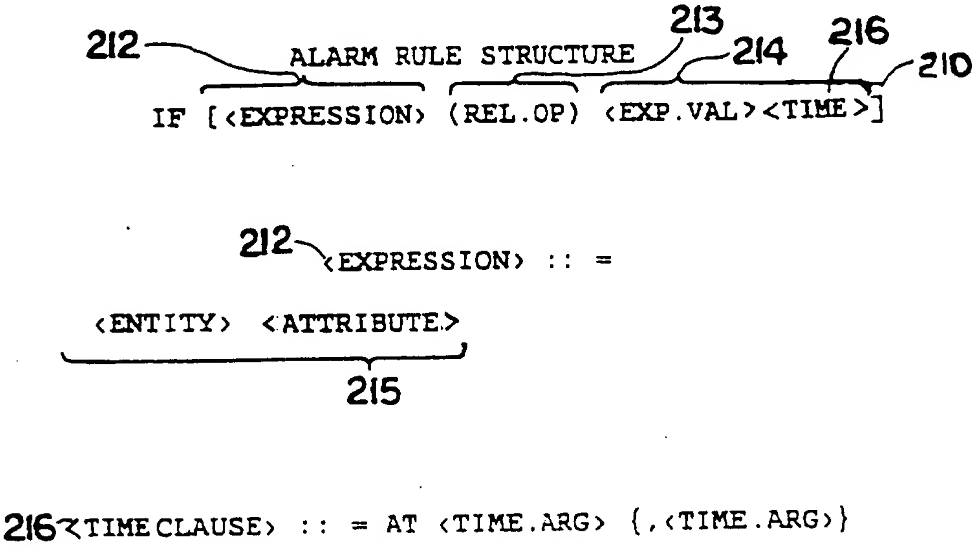


FIG. 10B

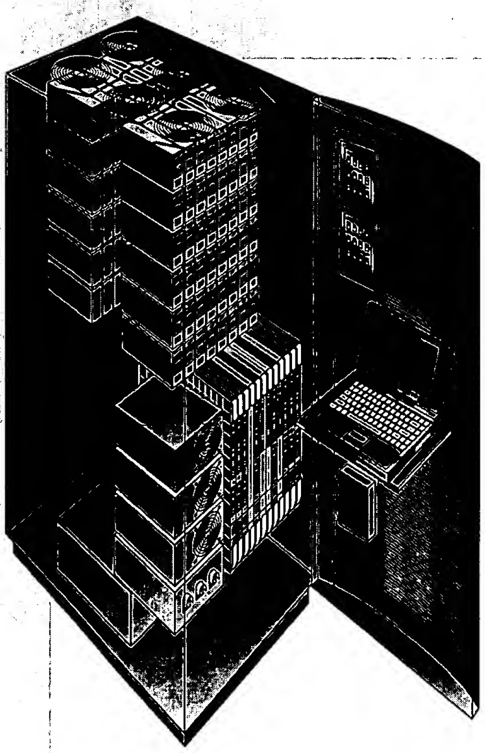

212
ALARM RULE STRUCTURE
213
214
216
210
IF [<EXPRESSION> (REL.OP) <EXP.VAL><TIME>]
212
<EXPRESSION> : : =
<ENTITY> <ATTRIBUTE>
215
216
<TIME CLAUSE> : : = AT <TIME.ARG> {,<TIME.ARG>}

EMC ENTERPRISE STORAGE SYSTEM

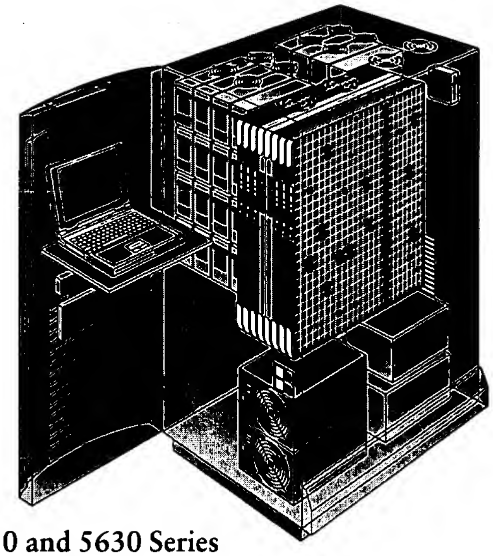
AR 1

EMC^2
The Enterprise Storage Company

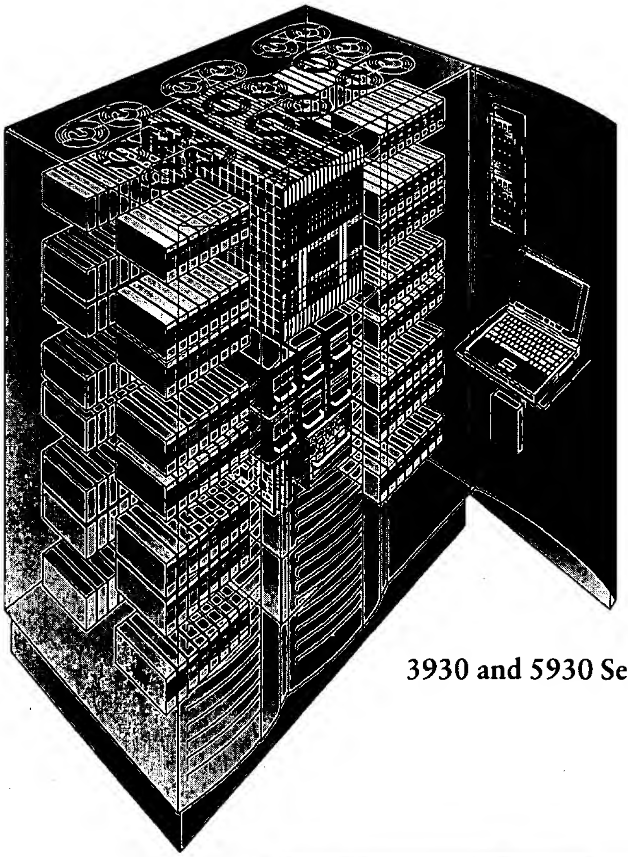
Symmetrix 3000 and 5000 Enterprise Storage Systems Product Description Guide



3830 and 5830 Series



3630 and 5630 Series



3930 and 5930 Series

Symmetrix 3000 and 5000 Enterprise Storage Systems Product Description Guide

Table of Contents

Chapter 1
Introduction

Overview

This technical overview provides information on the Symmetrix® 3000 and 5000 families of EMC Enterprise Storage™ systems, including product descriptions and details on key features and operations. This overview describes EMC's underlying storage system architectural philosophy which is based on the complementary MOSAIC:2000® hardware and ISA software architectures. The objective is to provide IS management and staff with a thorough technical understanding of Symmetrix systems.

There are currently three series in both the Symmetrix 3000 and 5000 families — the Symmetrix 3630, 3830 and 3930, and the Symmetrix 5630, 5830 and 5930. They form scalable families with leadership performance and capabilities in each of their respective capacity classes.

The Value of Symmetrix 3000 and 5000 Enterprise Storage Systems

Performance and Revenue Advantages

Symmetrix 3000 and 5000 systems help EMC customers enhance the performance and revenue of their businesses. These advantages typically can take any combination of three forms.

❶ Business Impact ❷ Operational Impact ❸ Financial Impact

Certain key features and capabilities of Symmetrix Enterprise Storage systems can contribute directly to achieving these impacts.

Business Impact usually provides the greatest customer value and payoff, and is most often a direct result of the industry-leading performance of Symmetrix systems. More transactions can be handled per hour, or less powerful servers can accomplish the task, because Symmetrix offers:

- **higher transaction throughput**
- **improved availability to information**
- **faster data analysis for decision support**

Higher performance can result in:

- **helping customers improve time-to-market of their products and services**
- **reducing development time for new business applications to create sources of new revenue**
- **enabling implementation of global services through improved data processing procedures**

Operational Impact is derived from fast, easy connectivity and integration, continuous data availability, and compatibility with existing technology. EMC customers can deploy their most valuable asset – people – to more productive efforts than having to manage data. Operational impact is typically associated with cost avoidance.

- **Continuous data availability and business continuance keep your business running**
  - Redundant critical components
  - Nondisruptive upgrades and repair of critical components
  - Data protection optimizes performance, availability, and price
  - Continuous data availability during migration from older technology disk systems to new Symmetrix systems
- **Compatibility and ease of implementation**
  - Support for all major multivendor servers: IBM®/PCM mainframes, heterogeneous UNIX® servers, PC LAN and Windows NT® servers, and AS/400® systems
  - Online host-independent mirroring between physically separated Symmetrix systems through Symmetrix Remote Data Facility (SRDF™), enabling business continuity during planned and unplanned outages

Financial Impact is often the most obvious and is frequently associated with both the ability to extend the useful life of technology, and the maintenance and data center environmental savings typically realized with Symmetrix products. EMC's hardware and software architectural approach to storage system implementation permits the seamless integration of new technology and capability as advances are made. Financial impact is typically associated with cost savings.

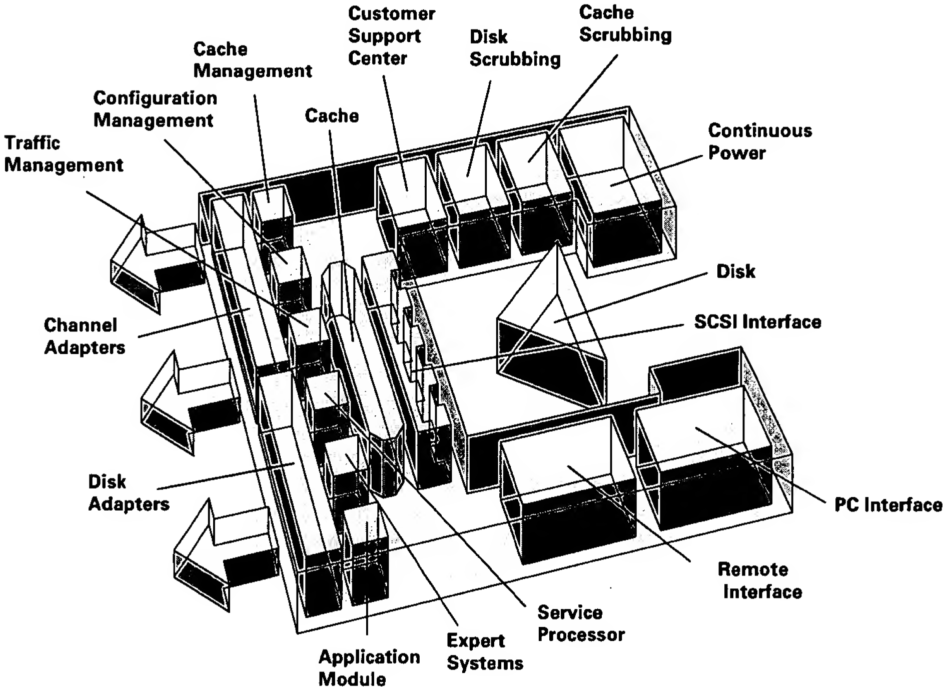
- **Asset protection**
  - MOSAIC:2000 is a modular hardware framework that allows rapid development and deployment of new storage technology while protecting existing investments.
  - ISA (Intelligent Storage Architecture) is a modular software framework that bridges the gaps between platforms, networks, databases, and applications. ISA also adds value to the storage investment with software such as SRDF, EMC InfoMover™, Symmetrix Data Migration Services (SDMS™), Symmetrix Manager, DataReach™, EMC TimeFinder™, EMC PowerPath™, EMC Data Manager, and the FDR family of backup/restore solutions.
- **Superior economic value**
  - Less floor space
  - Lower power and cooling requirements
  - Lower maintenance costs

EMC's Architectures for Enterprise Storage: MOSAIC:2000 and ISA

EMC Enterprise Storage systems make information protection, information sharing, and information management possible. As a shared central repository for valuable information, EMC Enterprise Storage:

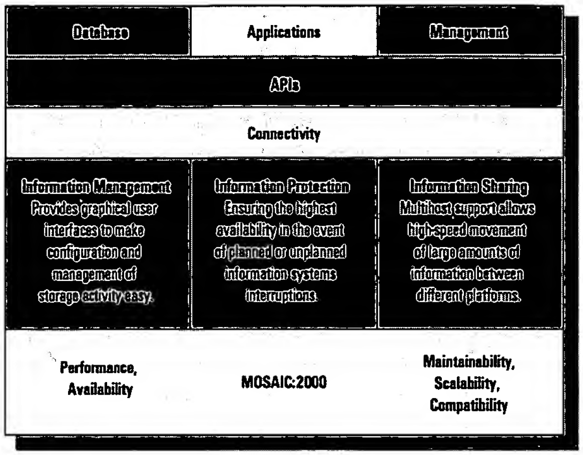
- Connects to disparate computer systems
- Provides common information management, protection and sharing capabilities
- Allows organizations to create a competitive advantage by leveraging large amounts of information

EMC Enterprise Storage systems rely on MOSAIC:2000 and Intelligent Storage Architecture (ISA) — a combination of industry-standard hardware and software. EMC Enterprise Storage architecture ensures optimum performance, availability, scalability and connectivity. Complementary MOSAIC:2000 hardware and ISA software architectures demonstrate the unique storage system philosophy of all EMC storage products and their capability to share information within an organization.



MOSAIC:2000 is the long-standing foundation for Symmetrix. This architecture combines industry-standard hardware with optimized software to provide the highest performance, availability, scalability and connectivity.

Intelligent Storage Architecture (ISA)


Database
Applications
Management
APIs
Connectivity
Information Management
Provides graphical user interfaces to make configuration and management of storage activity easy.
Information Protection
Ensuring the highest availability in the event of planned or unplanned information systems interruptions.
Information Sharing
Multihost support allows high-speed movement of large amounts of information between different platforms.
Performance, Availability
MOSAIC:2000
Maintainability, Scalability, Compatibility

ISA provides unique and powerful software that extends the capabilities of Symmetrix in the areas of information protection, information sharing and information management.

Together, MOSAIC:2000 and ISA yield powerful enterprise storage systems that:

- Provide high-level performance, capacity, and reliability
- Store and retrieve data from all major computing platforms, including mainframe and open systems environments
- Enable software-based functionality that ensures business continuance in the event of a disaster
- Deliver rapid and nondisruptive data migration from one system to another
- Share information, regardless of origin

EMC Storage Philosophy

- Highest performance, scalability, connectivity
- Information protection, information sharing and information management
- Provide intelligence at the storage system level
- Use industry-standard interfaces

Highest Performance

Symmetrix systems use large cache memory configurations, and EMC proprietary caching algorithms enable the highest probability for "cache hits" when reading data. One hundred percent cache fast writes ensure the highest possible performance when writing data. Fast, 100 percent cache writes enable Symmetrix performance to appear as close to that of solid-state disk as possible while being able to support the largest data capacity per system in the industry.

- *Scalability* – Symmetrix 3000 and 5000 systems enable consolidated storage strategies by providing scalable storage in a common family. System capacities scale from 35GB to multiple terabytes of fully protected storage. Symmetrix offers new ways to manage change and growth in applications, databases, servers, and overall business requirements.

- *Connectivity* – Connectivity to host platforms is provided through industry-standard interfaces. The Symmetrix 5000 series supports mainframe connections through ESCON®, parallel or block multiplexor channels. When optional Symmetrix ESP (Enterprise Storage Platform) software is installed, Symmetrix 5000 systems can simultaneously support open UNIX, Windows NT, and AS/400 systems with connectivity to fast-wide-differential (FWD) SCSI, Ultra SCSI, and Fibre Channels.

  Symmetrix 3000 systems support connectivity to open UNIX, Windows NT, and AS/400 systems with connectivity to FWD SCSI, Ultra SCSI, and Fibre Channels. When optional Symmetrix ESP software is added to Symmetrix 3000 systems, they simultaneously support mainframe connections through ESCON and block multiplexor channels.

  This level of Symmetrix connectivity enables simultaneous support of multiple hosts and multiple host types for greater configuration flexibility and the fulfillment of EMC's Enterprise Storage philosophy.

Information Protection

EMC software provides a variety of information protection/business continuance options, including Mirroring the optimum RAID level for both performance and availability.

The following software offerings supplement the EMC Enterprise Storage philosophy.

- Symmetrix Remote Data Facility (SRDF) provides fast enterprise-wide data recovery capability in the event of a planned or unplanned data center outage.
- EMC TimeFinder supports the online creation of multiple independently addressable business continuance volumes (BCVs) of information allowing other processes such as backup, batch, application development and testing, and database extractions and loads to be performed simultaneously with OLTP and other business operations.
- Symmetrix Dynamic Address Switching (S/DAS) offers the capability to dynamically swap DASD addresses in an SRDF environment that participates in a parallel sysplex environment. It also allows dynamic address swapping with Symmetrix Data Migration Services (SDMS).

EMC's Remote Support network can be used to upgrade Symmetrix operating software (microcode) on an operational Symmetrix system with minimal interruption of service. This unique approach upgrades Symmetrix software and functionality without downtime, combining fast functional enhancements with continuous data availability.

Information Sharing

Symmetrix provides centralized, sharable information storage that supports changing environments and mission-critical applications. This leading-edge technology begins with physical devices shared between heterogeneous operating environments and extends to specialized software that enhances information sharing between disparate platforms.

- EMC Celerra™ File Server enables direct attachment to networks for high speed centralized data storage and data sharing without the need for a general purpose server.
- Symmetrix Enterprise Storage Platform (ESP) software provides simultaneous mainframe and open systems support for Symmetrix 3000 and 5000 systems.
- Symmetrix provides standard simultaneous multiple open systems support.
- EMC InfoMover extends information sharing. InfoMover facilitates high-speed bulk file transfer between heterogeneous mainframe, UNIX, and Windows NT host platforms without the need for network resources.
- DataReach uses ESP as an enabling technology to provide access to mainframe database information, extract it, and transfer it to UNIX and Windows NT open systems relational databases.

Information Management

Symmetrix systems improve information management by allowing users to consolidate storage capacity for multiple hosts and servers. EMC offers powerful graphical user interface (GUI)-based tools that dramatically simplify and enhance Symmetrix configuration, performance, and status information gathering and management.

- Symmetrix Manager offers enhanced GUI-based storage monitoring, configuration, and performance tuning management capabilities for Symmetrix systems supporting open systems and mainframe environments.
- EMC PowerPath™ optionally offers a combination of simultaneous multiple path access, workload balancing, and path failover capabilities between Symmetrix systems and supported server hosts.
- EMC Volume Logix enables storage administrators to efficiently allocate Symmetrix storage in an Enterprise Storage Network environment to hundreds of UNIX or Windows NT servers located on the Fibre Channel hub or switched fabric.
- EMC Data Manager (EDM™) supports high performance network-based or directly connected open systems and Windows NT backup needs from one centrally managed site while offering a complete, high-performance database backup solution for the entire enterprise.
- Fast Dump/Restore (FDR) family of mainframe-based backup/restore utilities uses Symmetrix with existing mainframe infrastructures to provide a comprehensive suite of fast, nondisruptive information protection solutions for both mainframe and open systems environments.

Enterprise Storage Networks

Symmetrix systems can also serve as the central information repository in an EMC Enterprise Storage Network™ (ESN). An ESN provides an extremely high-speed, Fibre Channel-based network consisting of storage, switches, and hubs that expand EMC Enterprise Storage capabilities beyond the walls of the data center. ESN offers a fault-tolerant, self-healing enterprise wide architecture from which an enterprise can better manage, protect, and share all of its information resources.

System Intelligence

Traditional systems have placed the bulk of storage management decisions and overhead on the operating system and host processor. In this approach, decisions such as what data to cache and when to cache it take cycles away from applications and ultimately impact performance. Symmetrix systems can determine data access patterns in real time and intelligently optimize themselves for performance, independent of the host processor and operating system. Since these capabilities are not tied to specific operating systems or versions of operating systems, they can be exploited and do not require time-consuming and costly software upgrades. These capabilities are used for virtually all major mainframe, UNIX, Windows NT, PC LAN, and AS/400 systems without incurring host processor overhead.

Industry-Standard Interfaces

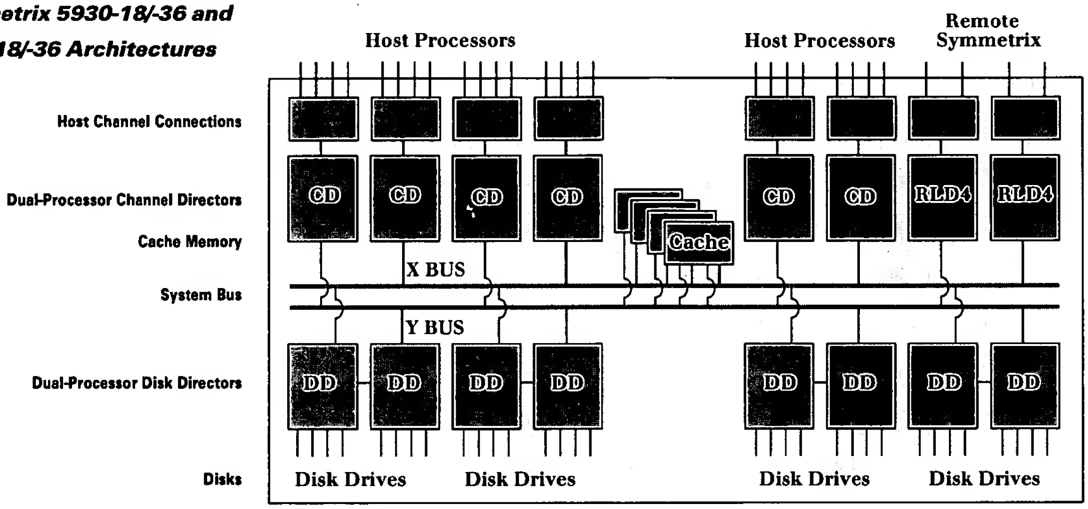
EMC uses open, industry-standard interfaces within Symmetrix systems. New, leading-edge HDAs can be introduced in the fastest possible manner, and popular open host processor interfaces are supported. This enables Symmetrix products to eliminate lagtime between the availability of new industry technology and new EMC deliverable products.

Chapter 2
Symmetrix 3000 and 5000 Enterprise Storage Product Description

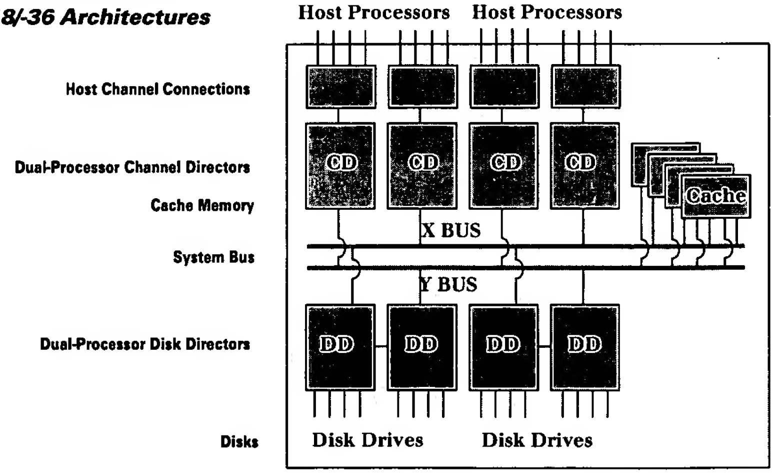
Symmetrix Enterprise Storage Family Configurations

Symmetrix systems support a mix of Channel Directors that include combinations of ESCON channels, parallel channels (Symmetrix 5000 only), Fibre Channel, Ultra SCSI and FWD SCSI channels, and Remote Link Adapters (used with SRDF and SDMS). Channel Directors are always installed in pairs to allow for redundancy and continuous availability in the event of repair or replacement to any single Channel Director.

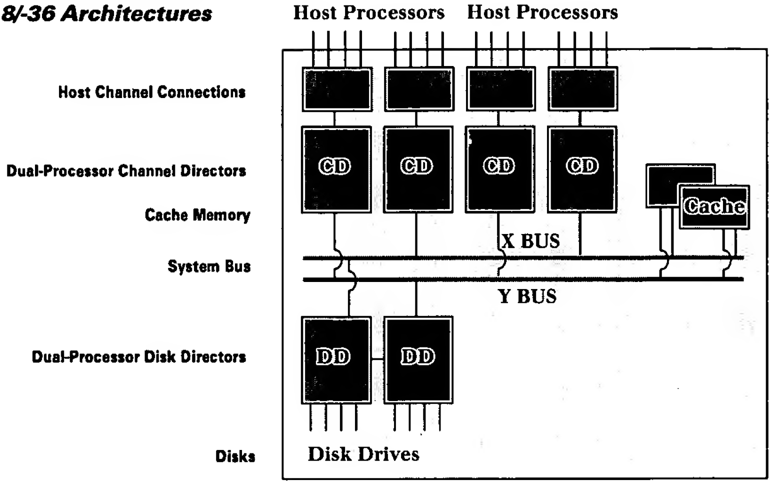
Symmetrix 5930-18/-36 and 3930-18/-36 Architectures



Symmetrix 5830-18/-36 and 3830-18/-36 Architectures

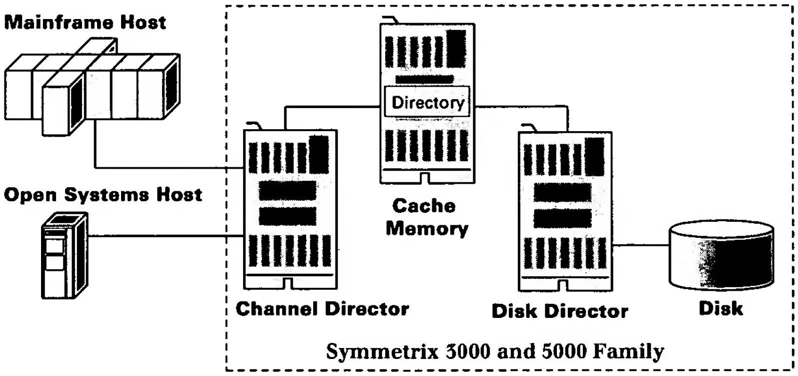


Symmetrix 5630-18/-36 and 3630-18/-36 Architectures



Basic Operation

Basic operations in the Symmetrix 3000 and 5000 family systems include Channel Directors, cache memory, Disk Directors, disks and the flow of data among these components, as illustrated in the following diagram.



Host Integration

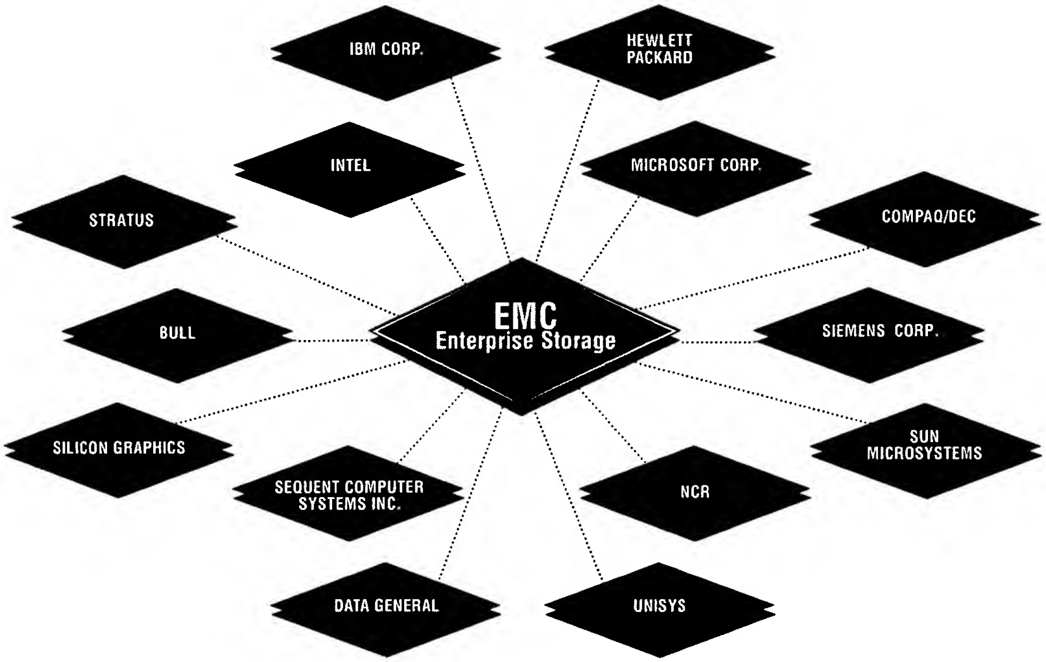
Symmetrix systems can be integrated easily and quickly with existing host processors, including all major enterprise servers and mainframes.

Multihost Support

Open systems connect to Symmetrix systems through fast-wide-differential (FWD) SCSI, Ultra SCSI, and Fibre Channel interfaces. The following list and illustration cover vendors of servers currently supported by Symmetrix systems. Support for additional system environments and additional system connections will continually be added, based upon customer demand and industry direction.

Bull®
Compaq®/DEC®
Data General®
Hewlett Packard®
IBM®
Intel®-Based Servers
NCR®
Sequent®
Siemens®
Silicon Graphics®
Stratus®
Sun Microsystems®
Unisys®

For a more detailed list of specific server models and supported operating system versions and interface technologies, contact your EMC representative or check EMC's web site at: *www.EMC.com/products/enterprise_storage_systems/open_sys_matrix.htm.*



Mainframe Operating System Support

In IBM®/PCM mainframe environments, all Symmetrix 5000 and 3000 with ESP systems are operating-system independent. The caching algorithms are self-managed and Symmetrix 5000 systems do not depend on host cache commands to receive the benefits of read and write caching. This means that the Symmetrix 5000 system will provide high performance and high functionality for I/O processing, not only to the latest ESA versions of mainframe operating systems but also to nontraditional mainframe operating systems and noncurrent versions of MVS, VM, and VSE. Virtually every System/370 and System/390® operating system can be supported, including:

MVS/ESA™
VM/ESA™
VSE/ESA™
MVS/XA™
VM/XA™
VSE/SP™
MVS/SP™
VM/SP™
MVT/VSE™
ACP/TPF™
VM/HPO™
AIX/ESA™

In addition, Symmetrix systems support other mainframe operating systems, including:

UTS® OS/1100® GCOS7™ GCOS8™ PICK™

Host Cluster Matrix

Configuring hosts in clusters achieves high availability and high performance. Cluster nodes share access to Symmetrix systems via fast-wide SCSI, Ultra SCSI, and Fibre Channel interfaces. Support for additional system environments and additional system connections will continually be added, based upon customer demand and industry direction. Symmetrix currently supports cluster hosts from the following vendors:

Bull
Compaq/DEC
Hewlett Packard
IBM
Intel-Based Servers
Sequent
Siemens
Sun Microsystems

For more detailed information on Symmetrix support for clustered environments, contact your EMC representative or check EMC's web site at:
www.EMC.com/products/enterprise_storage_systems/cluster_host_matrix.htm.

Device Support and Emulation

Symmetrix 5000 and 3000 systems with ESP appear to mainframe operating systems as a 3990-6, 3990-3 or 3990-2. The physical storage devices can appear to the mainframe operating system as a mix of multiple 3380 and 3390 devices. All models of the 3380 or 3390 volumes can be emulated up to the physical volume sizes installed. A single Symmetrix system can simultaneously support both 3380 and 3390 device emulations.

The Symmetrix responds to cache commands from the host processor and will respond as 3990-3 or 3990-6, but will not always perform the command in exactly the same manner as 3990-3 or 3990-6. Some host access methods are designed to turn off cache during sequential processing. This is necessary with conventional cached controllers as their caching algorithms create cache pollution when processing sequential I/O. The sequential prefetch capability of Symmetrix allows for efficient sequential operation without having to actually turn off Symmetrix cache. This allows the Symmetrix to provide the high performance of an integrated cached environment 100 percent of the time, while the host operating system perceives that cache has been turned off.

The Symmetrix emulation of the IBM 3990-3 or 3990-6 allows it to be compatible with IBM's Systems Managed Storage (SMS) and other data management systems. Symmetrix knows how data is being accessed and will manage its own caching and prefetch processes accordingly. EMC cache management algorithms select which channel commands to process and which to ignore for greater efficiency and performance.

On open systems hosts, Symmetrix logical disk volumes appear to the host as physical disk devices at SCSI target ID/logical unit number addresses. All host logical volume manager software can be used with Symmetrix disk volumes.

When using a FWD SCSI connection to an open system processor, the Symmetrix system appears as industry-standard SCSI disk devices behind a FWD SCSI interface and data is stored in Fixed Block Architecture (FBA) format.

Host Channel Connection

All Symmetrix systems provide exceptional channel connectivity through combinations of Channel Directors. These include ESCON channels, parallel channels (5000 systems only), FWD SCSI and Ultra SCSI channels, Fibre Channels, and Remote Link Adapters (used with SRDF and SDMS). Channel Directors are installed in pairs, providing redundancy and continuous availability in the event of repair or replacement to any one Channel Director.

Channel Directors

Symmetrix systems support mainframe, UNIX, Windows NT, and AS/400 connections through Channel Directors. They connect directly to host processors through physical path types or Physical Channel Attachments.

The Symmetrix 3000 and Symmetrix 5000 family systems with ESP support open UNIX systems, Windows NT systems, and AS/400 connectivity through Symmetrix FWD SCSI, Ultra SCSI, and Fibre Channel Channel Directors. Each Channel Director is a single board with four host connections (two for Fibre Channel) and, depending on the Symmetrix, from two to eight Channel Directors.

The Symmetrix 5000 and Symmetrix 3000 family systems with ESP support mainframe connectivity through serial Channel Directors for ESCON connections and parallel Channel Directors (Symmetrix 5000 systems only) for block multiplexor connections. Each channel connection supports four channel connections. For configuration flexibility, these directors can be installed simultaneously and, depending on the Symmetrix, from two to eight Channel Directors are supported.

When Symmetrix ESP software is installed on a Symmetrix 5000 or 3000 system, simultaneous connections for mainframes, UNIX, Windows NT, and AS/400 systems are provided. This specialized software enables combinations of serial Channel Directors, parallel Channel Directors, FWD SCSI Channel Directors, Ultra SCSI Channel Directors, and Fibre Channel Directors on the same Symmetrix system.

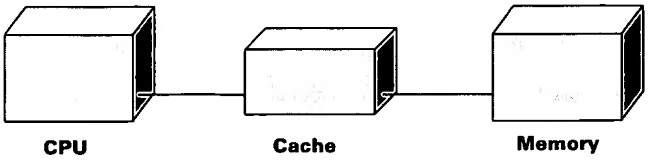
Internal Data Flow

Data Flow

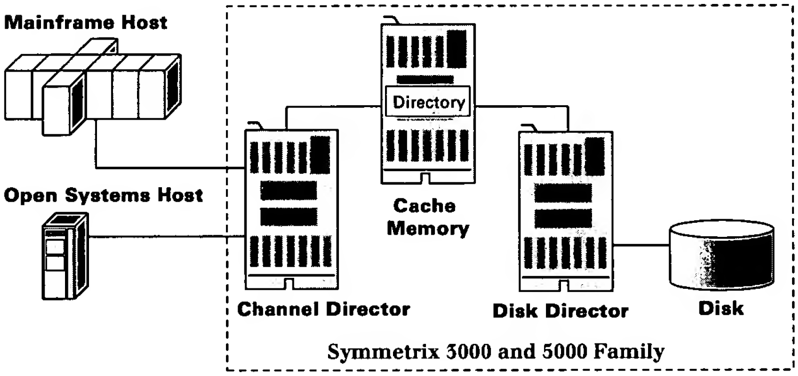
Intelligent cache configurations allow Symmetrix systems to transfer data at electronic memory speeds which are much faster than physical disk speeds. Symmetrix products are based on the principle that the working set of data at any given time is relatively small when compared to the total system storage capacity. When this working set of data is in cache, there is a significant improvement in I/O performance. The performance improvement achieved is dependent on both:

- *Locality of Reference* – If a given piece of data is used, there is a high probability that a nearby piece of data will be used shortly thereafter,
- *Data Reuse* – If a given piece of data is used, there is a high probability that it will be reused shortly thereafter.

This cache principle has been in use for years on host processor systems (CPU and storage devices). The figure below illustrates this type of host cache use. The cache used in this manner is often a high-speed, high-cost storage unit used as an intermediary between the CPU and main storage.



Symmetrix uses the same cache principle as host systems, but with enhanced caching techniques. The following diagram illustrates cache use in Symmetrix.

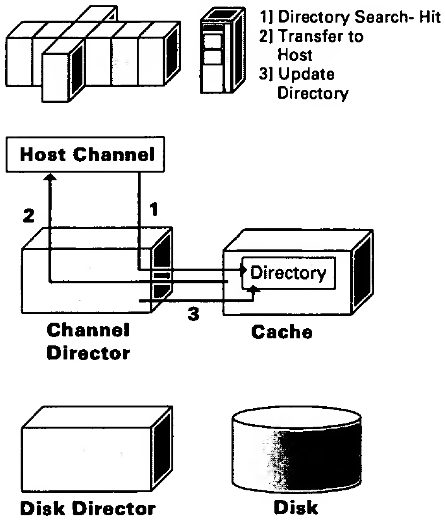


In the Symmetrix system, the Directors perform the following functions:

- Each Channel Director handles I/O requests from the host. It accesses the directory in cache to determine if the request can be satisfied within the cache. The directory contains information on each cache page and blocks within each page.
- Each Channel Director manages cache using an Age Link Chain table and Least Recently Used (LRU) algorithm. An Age Link Chain table maintains the references to the Most Recently Used (MRU) to Least Recently Used page locations. The LRU algorithms use the information in this table to ensure that only pages of data that have been used recently are kept in cache.
- A prefetch algorithm dynamically detects sequential data access patterns to the disk devices. The directors improve the hit ratio of these accesses by promoting blocks from the disk devices to cache slots before that data has been requested. The prefetch algorithm can stage two to 12 tracks to cache depending on access patterns learned.
- The Disk Director manages access to the disk drives. It performs a background operation that destages "written-to" blocks to disk.

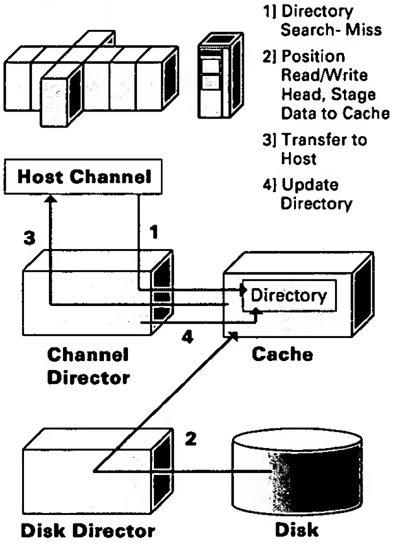
Four basic types of operations occur in a Symmetrix system: Read Hit, Read Miss, Fast Write, and Delayed Fast Write operations. The following diagrams illustrate these operations.

Read Hit



1] Directory Search- Hit
2] Transfer to Host
3] Update Directory

Read Miss

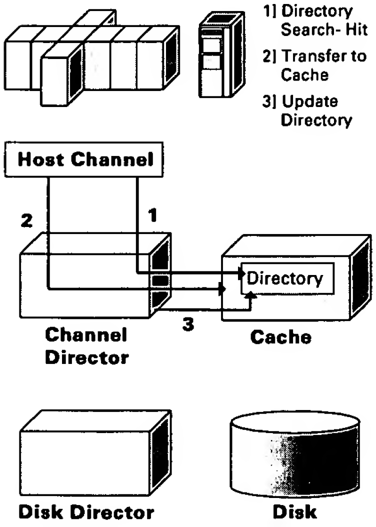


1] Directory Search- Miss
2] Position Read/Write Head, Stage Data to Cache
3] Transfer to Host
4] Update Directory

A *Read Hit* occurs on a read operation when all data necessary to satisfy the host I/O request is in cache. The Channel Director transfers the requested data from cache to the host and updates the cache directory.

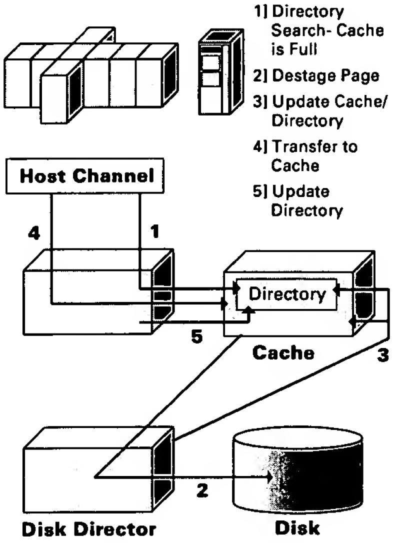
A *Read Miss* occurs when all data necessary to satisfy the host I/O request is not in cache. The Disk Director stages the block(s) containing the data from disk. The Disk Director places the block(s) in a cache page. Simultaneously, the Channel Director reconnects to the host and sends the requested data.

Fast Write Hit



1] Directory Search- Hit
2] Transfer to Cache
3] Update Directory

Delayed Fast Write



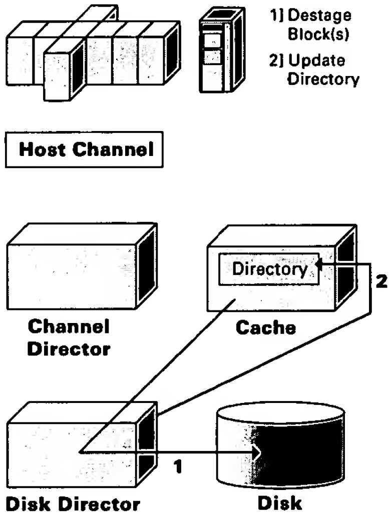
1] Directory Search- Cache is Full
2] Destage Page
3] Update Cache/ Directory
4] Transfer to Cache
5] Update Directory

A *Fast Write* occurs when the percentage of modified data in cache is less than the Fast Write threshold. On a host write command, the Channel Director places the incoming block(s) in cache.

A *Delayed Fast Write* occurs only when the Fast Write threshold has been exceeded (that is, the percentage of cache containing modified data is higher than the Fast Write threshold). The Least Recently Used data is destaged to disk. When sufficient cache space is available, the Channel Director processes the host I/O request as a Fast Write. With sufficient cache present, this type of cache operation will rarely occur.

A background operation also occurs in Symmetrix systems. This background operation destages "written-to" blocks to disk. This allows any written-to or changed data to be maintained in two locations: cache for high performance in the occurrence of reuse of that data, and on disk to maintain the highest levels of data integrity. All pending writes are assured of arrival to the intended disk even in the event of power failure. The following diagram illustrates this destaging operation.

Destaging Operation



Cache Memory

One of the most crucial components of a Symmetrix system is the cache memory. All read or write operations transfer data to or from cache. Any transfers between the host processor, Channel Directors, and cache are achieved at electronic speeds that are a quantum leap faster than transfers involving disk. Optimization around the movement of data between disk and cache results in the highest performance possible. There are two cache buses, x and y; each has a 360MB per second bandwidth for a total processing bandwidth of 720MB per second.

Performance Features

Performance remains a significant differentiation between Symmetrix systems and all alternative disk offerings. The features that will be discussed all impact performance and contribute to increasing transaction volumes, improving online response time, and reducing the time needed to execute batch runs.

Electronic Data Transfer

Symmetrix greatly exceeds the throughput and response time performance of conventional disk storage because the majority of data is transferred at electronic memory speeds, not at the dramatically slower speeds of physical disk devices. The Symmetrix system's intelligent use of up to 16GB of cache contributes greatly to this performance advantage.

Advanced Caching Algorithms

Simply having these robust cache configurations is not enough. One of the fundamental differences between Symmetrix products and all other DASD is the advanced caching algorithms that allow intelligent usage of the installed cache for high performance. These algorithms search quickly and efficiently to determine whether the requested data is in cache. They also understand how the application is accessing the data and tune themselves accordingly in real time. The cache management algorithms respond to channel requests to manage the cache via host processor software when appropriate and perform the management functions independently when the host processor does not make requests. This is a complex series of tasks and requires the advanced cache management algorithms of Symmetrix to accomplish them effectively.

With the large amounts of cache offered on Symmetrix systems, the typical installation will attain a read hit ratio (requested data is in cache) of 90 percent to 95 percent. In some alternative cached environments, read performance may be acceptable due to the opportunity of a read hit, but write performance is inferior unless there are DASD Fast Writes (DFW). Even with DFW, these alternative products often have a very limited resource, Nonvolatile Storage (NVS), for this function. In the event of a power outage, NVS is only nonvolatile for the cache and only for a limited period of time. The Symmetrix systems, however, always provide 100 percent system nonvolatility, allowing all writes to be "fast writes." Channel End/Device End is presented to the host channel when the data is written to cache and verified.

Efficient Cache Searching

One of the problems of traditional controllers with large cache configurations is the lack of an ability to search the cache in an efficient manner. Increasing cache configurations means that the search time increases proportionally. This search time is added to every I/O request, read hit, write hit, read miss, or write miss. This is a considerable penalty for every I/O request, especially in performance-critical applications. The controller may actually disconnect from the channel during this process and must then reconnect if there is a cache hit.

The Symmetrix systems perform the cache search via advanced proprietary algorithms. It only requires 20 microseconds to determine if a record is in cache. This advanced algorithm allows the 20 microsecond search time to remain constant regardless of cache configuration. With a 20 microsecond cache search, there is no reason to disconnect from the channel during the search. In fact, it takes longer to disconnect and reconnect than it does to perform the cache search. In normal operation, the only time that a Symmetrix system will disconnect from the channel is in the case of a read miss.

Sequential Prefetch

Symmetrix systems continually monitor I/O activity and look for access patterns. When the second sequential I/O to a track occurs, the sequential prefetch process is invoked and the next track of data is read into cache. The intent of this process is to avoid a read miss. Once the first track is completely read by the host processor, the third track is read and reuses the same cache location as the first.

This process of using the track slots in a round-robin fashion prevents cache pollution caused by conventional sequential caching algorithms. Should a read miss occur, the Symmetrix system will increase the number of track slots from two to five. If a read miss still occurs, the Symmetrix prefetch routines will increase the slots to eight. The maximum number of track slots that will be allocated for a sequential operation is 12. Should I/O activity reduce, the number of track slots will be reduced accordingly. When the host processor returns to a random I/O pattern, the Symmetrix system will discontinue the sequential prefetch process.

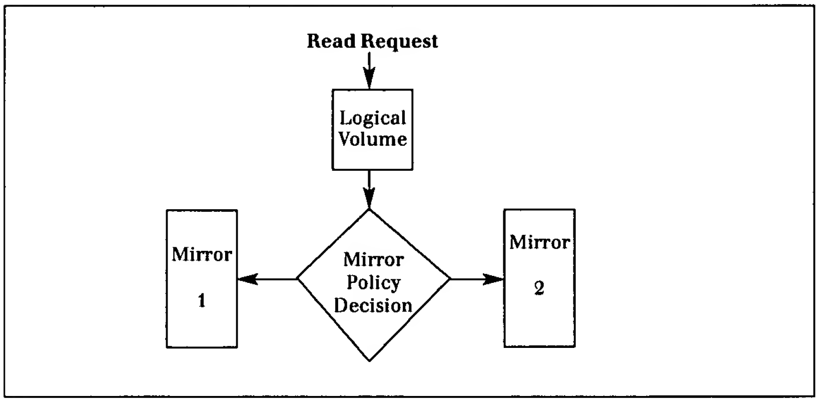
PermaCache Option

Symmetrix allows you to permanently assign mission-critical data requiring extremely high performance to cache. A variable number of contiguous cylinders on the disk devices can be reserved for PermaCache backup.

PermaCache is best used for infrequently accessed data that needs instantaneous response since this data normally may not be in cache at the time it is requested. The large cache and intelligent caching algorithms strive to keep frequently accessed data in cache, making its assignment to PermaCache unnecessary. PermaCache requires additional cache memory to be available above the base cache required for any particular configuration.

Dynamic Mirror Service Policy

Symmetrix Dynamic Mirror Service Policy (DMSP) is an enhancement to Symmetrix which provides the algorithms for processing read operations for mirrored (RAID 1) volumes. As shown, DMSP determines which mirrored volume will service each read request. Using only a static mirror service policy results in a single mirror always being used and no performance advantage when using mirrored volumes over non-mirrored volumes.



The DMSP feature takes advantage of static mirror service policies, but addresses their limitations by making periodic adjustments. Over time, data access pattern information is collected and results in a decision about the best policy to use for each volume. Volumes with higher access rates get more preferential treatment than volumes with lower access rates. The result is improved overall system performance and reduction or elimination of arduous studies of access patterns and manual configuration changes.

Disk Drives

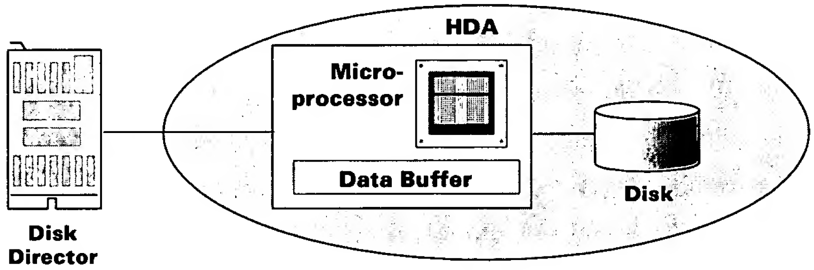
EMC uses advanced technology disk drives and disk controllers to enhance Symmetrix capabilities. Microprocessors embedded in the controllers of each drive enable capabilities such as advanced RAID protection to be offloaded from CPUs, disk directors and controllers. This further enhances the performance of Symmetrix systems and also serves to make them more easily compatible with a wide array of hardware and software platforms.

Symmetrix 3630/5630, 3830/5830, and 3930/5930 systems support mixed configurations of up to 32, 96, and 256 (respectively) 18GB and 36GB disks drives. This breadth of scalable capacity and configuration choices allows Symmetrix systems to adapt to virtually any enterprise storage requirement.

Disk Directors

The Disk Directors manage the interface to the physical disk, and are responsible for data movement between the HDAs and cache. HDAs are connected to Disk Directors through industry-standard SCSI interfaces with two microprocessors per Disk Director. This connection allows rapid introduction of the latest disk drive technology into Symmetrix systems.

Disks



Symmetrix systems use industry-standard SCSI HDAs for physical disks. The use of industry-standard HDAs allows EMC to keep pace with customer needs as technology advances in the area of increased capacities and improved performance. Each HDA is configured with its own controller consisting of control logic, a microprocessor, and a device-level buffer. The device-level buffer is designed to eliminate Rotational Position Sensing (RPS) misses. An RPS miss occurs when the head or current rotational position of the disk media is such that the transfer is possible when requested, but the controller and its path to an HDA are not available. An RPS miss typically causes a time delay for transfer of at least one additional rotation of the disk. Through the use of the device-level buffer, data is easily moved between the Disk Director or the drive media and the buffer. This enables an electronic transfer between the buffer on the HDA and the Disk Director, thereby avoiding the possibility of an RPS miss. It can also be segmented, using the SCSI-2 command set. This allows Symmetrix control logic to issue a read command to the device, detach, then issue a data write to a different segment of the device-level buffer. The onboard microprocessor will manage the read or write, and will notify the Symmetrix system when the read is complete.

Every HDA contains its own microprocessor which has the capability of self-management. This gives Symmetrix the ability to perform parallel tasks, such as diagnosis and simultaneous transfers, and further enhances performance.

Hyper-Volume Extension

Symmetrix enhances disk system functionality by supporting up to 32 logical volumes on one physical device. Up to a maximum of 4,096 logical volumes are supported on a Symmetrix system.

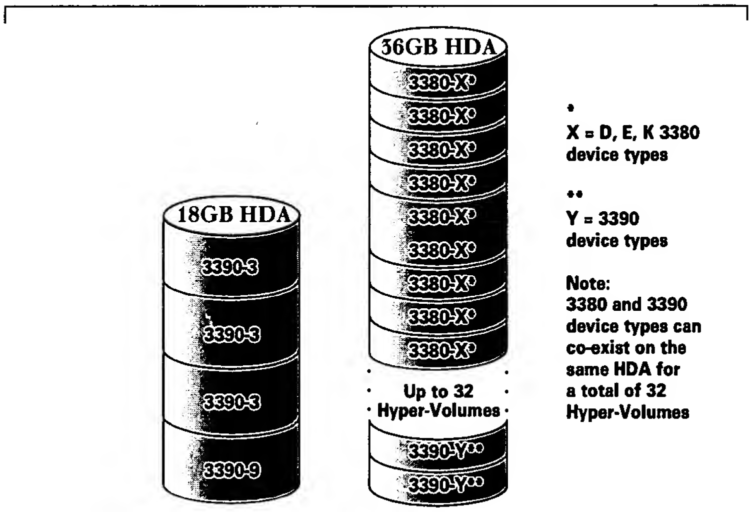
For mainframe customers, there are two separate uses of Hyper-Volume Extension (HVE).

- *Extended Cylinder Addressing for higher performance* – Beyond the capacity on the drive required for IBM device emulation, there can be an additional small logical volume for data sets that require very high performance (Multi-Image Manager files, JES Checkpoint, RACF Control files, etc.). Since this small logical volume is separate from other volumes, Unit Control Block (UCB) busy conditions due to contention are eliminated.
- *Split-Volume Capability for greater flexibility* – Up to 32 separate logical volumes can be configured on a single physical drive. For example, a single EMC 18GB drive could support up to six logical 3390-3s, or up to 18 logical 3390-1s, or nine logical 3390-2s or two logical 3390-9s. A single 36GB drive could support 12 logical 3390-3s, or up to 32 logical 3390-1s, or 18 logical 3390-2s, or four logical 3390-9s. This flexibility provides for the consolidation of many physical DASD devices into far fewer physical high capacity, high performance disks. HVE enables the replacement of older storage devices without requiring application or data format changes.

Support is provided for native IBM 3390 and 3380 track emulation with all 3390 and 3380 disk volumes being supported. No modifications are required to the operating system, application, or program product software to take advantage of HVE.

The following illustration represents some examples of possible Hyper-Volume Extension logical configurations of EMC 18GB and 36GB drives in mainframe environments.

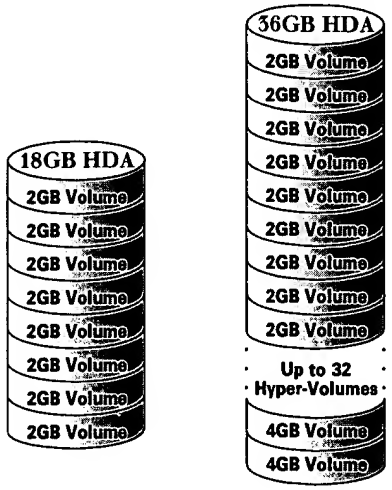
Mainframe environments with up to 32 logical volumes per disk



Open systems customers can also take advantage of Hyper-Volume Extension to support multiple logical disk volumes on individual disks. This capability is particularly useful for some 32-bit implementations of UNIX that allow only 2GB file systems per single logical disk. For example, up to eight 2GB logical disk volumes can be defined for a single EMC 18GB HDA.

The following illustration represents examples of possible Hyper-Volume Extension logical configurations of EMC 18GB and 36GB drives in open systems environments.

Open systems environment with up to 32 logical volumes per disk



Note:
Different volume capacities can co-exist on the same HDA for a total of 32 Hyper-Volumes.

Meta Volume Addressing

Symmetrix also enhances disk system functionality in Windows NT and open systems environments through the capability of meta volume addressing. Symmetrix allows the concatenation of contiguous logical devices, up to a maximum of 512GB per meta device. This overcomes the addressing limitations imposed in Windows NT environments.

Symmetrix Data Protection

EMC has chosen to enhance the basic RAID level definitions in each of the two implementations of data protection that are offered for Symmetrix. Unique customer value can be derived from the ability to have Symmetrix products support disk arrays that can be protected with:

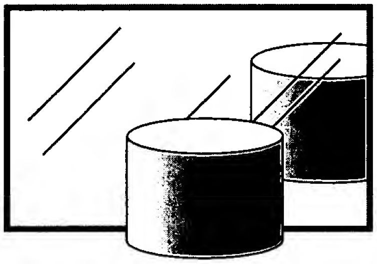
- *Mirroring (RAID 1)* – for mission-critical data with highest performance and highest availability.
- *Symmetrix Remote Data Facility (SRDF)* – an enhanced version of mirroring for multiple storage system data protection and availability that can include multiple sites.

This capability allows optimization for the best relationships of availability, performance, and cost for individual data sets. These options are configurable at the physical volume level so that different levels of protection can be applied to different data sets within the same Symmetrix system. This unique flexibility allows the customer to maintain the lowest possible costs in relation to the necessary levels of performance and data availability.

The EMC Symmetrix implementations of data protection are able to exploit Symmetrix functionality that differentiates the EMC offerings from typical RAID offerings as follows.

Mirroring (RAID 1)

The implementation of RAID 1 Mirroring on Symmetrix systems includes performance enhancements beyond the high availability capabilities normally associated with RAID 1.



Write Operations with Mirroring

A write operation to any mirrored volume is executed identically to a nonmirrored write. The Channel Director presents Channel End/Device End to the host after data is written and verified in cache. The Disk Directors then destage the data to each drive of the mirrored pair of drives asynchronously. As such, Mirroring on Symmetrix exploits the 100 percent fast write capability, and the application does not see additional time associated with having to physically perform two disk write I/Os (one to each drive of the mirrored pair) as is normally associated with RAID 1.

Read Operations with Mirroring

The Symmetrix performance algorithms for read operations in mirrored pairs offer three service policies to best balance the use of the Symmetrix architecture. Interleave Service Policy shares the read operations of the mirrored pair by reading tracks from both HDAs in a flip flop method, a number of tracks from M1, and a number of tracks from M2. Interleave is designed to achieve maximum throughput. Split Service Policy differs from Interleave because read operations are assigned to either the M1 or the M2, but not both. In the case of multiple hyper-volumes in the mirrored pair, certain logical volumes are read exclusively from M1 and certain logical volumes are read exclusively from M2. Split is designed to minimize head movement Dynamic Mirrored Service Policy (DMSP) utilizes both Interleave and Split for maximum throughput and minimal head movement. DMSP adjusts each logical volume dynamically based on access patterns detected.

Mirroring Error Recovery

In the unlikely event that one disk in the mirrored pair fails, the Symmetrix automatically uses the other disk drive of the mirrored pair without interruption of data availability. The Symmetrix system notifies the host operating system of the error via the message to operator protocol and to the EMC Customer Support Center via an Auto-Call action. The EMC Customer Support Center Product Support Engineer (PSE) then begins the diagnostic process and, if necessary, dispatches a Customer Engineer (CE) to the customer site. Once the suspect HDA is nondisruptively replaced, the Symmetrix system re-establishes the mirrored pair and automatically resynchronizes the data with the new disk. During the data resynchronization process, the Symmetrix system gives priority to host I/O requests over the copy I/O to minimize the impact on performance.

Mirroring Advantages

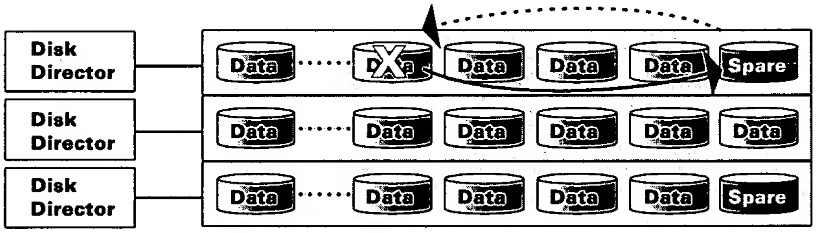
In summary, Mirroring provides:

- Improved performance over traditional RAID 1 by supporting 100 percent fast write, and two simultaneous internal data transfer paths.
- Protection of mission-critical data from any single point of failure.
- Continuous business operation by switching to the alternate HDA of a mirrored pair without interruption to data availability should loss of access occur to one of the HDAs in the mirrored pair.
- Assurance that the second copy is identical to the first copy.
- Automatic resynchronization of the mirrored pair after repair of the suspect volume.
- Transparency to the host processor and operating system.

It is possible to provide even greater protection for data that is already protected by mirroring by defining spare disks for disks protected by mirroring. In the event of the error threshold having been exceeded for a volume that is already protected by mirroring, instead of starting the dynamic sparing process of copying from the failing disk to the spare disk, data will be copied from the "good" disk to the spare disk. This will provide additional protection for the remaining active disk of the mirrored pair in case the failing disk cannot be immediately replaced.

Dynamic Sparing

Symmetrix family systems can provide Dynamic Sparing, an additional level of protection for volumes that use the mission-critical redundancy of RAID 1 Mirroring. A small pool of spare volumes is committed to this option, typically in groups of spares equal to the number of data volumes in the RAID 1 group. This user-selectable option is capable of providing dynamic reallocation of data to a standby spare, thus maintaining data protection in the event of an HDA failure.



Synchronization and HDA Replacement

Once all HDAs in the RAID group are synchronized, the spares become the active volumes, the failing HDA is taken offline, and notification is made of the occurrence. Notification is made to the host via sense information and to the EMC Customer Support Center via an Auto-Call event. The local Customer Engineer is notified and will then report onsite to perform a nondisruptive replacement of the reported failing HDA. Once the physical replacement is complete, microcode is notified and the new HDA is synchronized with the Dynamic Spares in use during this process. Because data volumes are fully protected, the HDA replacement and resynchronization can be deferred to a time convenient to the customer. When the synchronization is complete, the HDA in the original location becomes the operational HDA, leaving the spares standing by and ready should another HDA fail at some time in the future. Throughout this process, continuous data availability is provided to users and applications without any disruption.

Dynamic Sparing Operation

The entire Dynamic Sparing process requires no intervention from customer personnel as it is completely implemented in Symmetrix microcode. All that is required from an operational perspective is to select the Dynamic Sparing option during initial Symmetrix system configuration and to reserve the necessary spare HDAs. Priority is given to host I/O requests during resynchronization, so high performance can be maintained even during asynchronous data resynchronization. Since errors are usually detected well in advance of an actual disk failure, dynamic sparing has proven itself to be very effective at being able to copy a full drive to a spare drive prior to data becoming unavailable on the failing drive.

Symmetrix Backup Restore Facility (SBRF)

Symmetrix 5000 systems have the ability to perform high-speed backup and restoration of data while applications remain online and accessible with the Symmetrix Backup Restore Facility (SBRF). SBRF is compatible with IBM's Concurrent Copy, but provides a much higher performance environment.

Service and Maintenance *Goals and Philosophy*

The goal for Symmetrix products is to be able to address all possible aspects of systems operation that contribute to providing continuous data availability to allow continuous business operation. The philosophy of EMC is to design in maximum reliability and then to implement the design with the most reliable components available. Once the design and component selection are complete, the reliability focus continues with Design Verification Testing (DVT), Highly Accelerated Life Testing (HALT), and Ongoing Reliability Testing (ORT) to assure customers of an inherently highly reliable product at all times. EMC also employs extensive leading-edge Environmental Stress Screening (ESS) techniques to weed out early life component failures well before the Symmetrix system is delivered to the customer site. Beyond the redundant hardware components and basic microcode operations of Symmetrix, EMC provides significant data protection and RAID offerings that provide continuous data availability in the event of disk failures or even the total loss of a data center.

Building upon this foundation of highly reliable components, the architecture of the Symmetrix focuses on redundancy so that data availability is assured even in the unlikely case of a component failure. In addition to redundant data paths, redundant components exist within all the major functional units, providing backup should a component failure occur.

The concept of continuous data availability is extended further to one of business continuance with capabilities offered by Symmetrix Remote Data Facility (SRDF), EMC TimeFinder, Symmetrix Data Migration Services (SDMS), and the FDR family of backup/ restore software products. These EMC-unique offerings provide continuous data availability and continuation of business operations in situations where alternative DASD would typically require multiple hours or days of application downtime.

Nonvolatile Power System

The entire Symmetrix system is made nonvolatile via an onboard battery backup system. The battery backup system provides the means for destaging any fast write data that might be in cache should an AC power failure occur. In addition to providing nonvolatility to the Symmetrix system, the batteries are fully capable of powering not only all electronic components, but also all HDAs during this time. This means that the disks are always powered down in an orderly manner, eliminating emergency power off situations and extending their useful life considerably.

The Symmetrix system battery will keep the entire system powered long enough to destage all write tracks currently in cache. Symmetrix will continue to accept host I/Os for a period of three minutes. If external power is not restored after three minutes, Symmetrix will return a Device Not Ready condition for all devices to all connected hosts. Symmetrix will then destage all write tracks currently awaiting destage and then perform an orderly shutdown. An orderly shutdown is a condition where the heads on the HDAs are properly retracted and the drives are spun down and powered off. Should AC power be restored prior to the Symmetrix being powered down, the Symmetrix becomes immediately operational without requiring a system restart.

The power system provides similar redundancy if a power supply fails. There is sufficient capacity in the remaining power supplies to maintain full operation until a nondisruptive repair can be made to the failed component. Additional redundancy is provided in the system backplane with two duplicate busses providing redundant data paths should a catastrophic type of failure occur in this component. Two fully redundant AC power lines are provided and if one power source is lost, the other will provide continuous operation.

Self-Maintenance and Continuous Data Availability

Symmetrix has full state-of-the-art self-monitoring, self-diagnosing, and, where possible, self-repairing algorithms. The objective of this philosophy is the avoidance of user-observable errors. Symmetrix will actively identify internal temporary errors that could potentially lead to any type of user-observable hard failure and attempt to correct them prior to data being unavailable to a user or an application. This error avoidance is accomplished through a process of error detection, error logging, and notification.

During idle time, the disks are read ("disk scrubbing"), looking for any type of error. Upon sensing a correctable error, the error is corrected and then rewritten. The block of data is read again to verify that it was a permanent correction. If it is correctable, the pertinent information is logged and scrubbing continues. If the error is not permanently corrected, the process is repeated until it is either corrected or the error recovery routines determine that a skip defect must be executed. If the skip defect must be executed, it is done via Symmetrix microcode. When the skip defect is complete, notification is made and the scrubbing process continues. Should a sufficient number of skip defects occur on a track that would make an alternate track assignment necessary, that too is accomplished through Symmetrix microcode and is transparent to the user.

"Cache scrubbing" is accomplished in a manner similar to disk scrubbing. During idle time, cache is checked for any single bit errors. Should a single bit error be encountered, it is corrected and the line of cache is rewritten and then read to determine if it was permanently corrected. If the single bit was permanently corrected, a counter is incremented, the error is logged, and processing continues. If the error was not permanently corrected the first time, the correction process continues until either the correction is permanent or microcode determines the single bit error is not correctable. Should it be determined that the single bit error is permanent, that section of cache is taken offline. This process of "fencing off" allows EMC to take the section of cache out of service prior to the customer seeing a temporary error.

Nondisruptive Component Repair

All Field Replaceable Units (FRU) of Symmetrix systems are capable of nondisruptive repair, including microcode and hardware. Intermediate versions of microcode are capable of being loaded without disruption to data availability for users and applications. Major hardware FRUs consist of the following:

- Channel Directors
- Disk Directors
- Head and Disk Assemblies (HDAs)
- Cache Memory Cards
- Power Supplies
- Battery System
- Fan Subsystems

Nondisruptive Microcode Upgrades

Microcode upgrades, performed by the EMC Product Support Engineers (PSEs) at the EMC Customer Support Center, provide enhancements to performance algorithms, error recovery and reporting techniques, diagnostics, and microcode fixes.

Nondisruptive microcode upgrades are available for Symmetrix systems. Symmetrix takes advantage of its multiprocessing and redundant architecture to allow for hot loadability of similar microcode platforms.

During a nondisruptive microcode upgrade, the PSE downloads the new microcode to the service processor. The new microcode loads into the EEPROM areas within the Channel and Disk Directors, and remains idle until requested for hot load into control storage. The Symmetrix system does not require manual intervention on the customer's part to perform this function. All Channel and Disk Directors remain in an online state to the host processor, thus maintaining application access. Symmetrix will load executable code as selected "windows of opportunity" within each director channel resource until all have been loaded. Once the executable code has been loaded, internal processing is synchronized and the new code becomes operational. This capability can be utilized to upgrade or to back down from a release level or interim update.

Data Integrity

Checking mechanisms are necessary throughout the data path within the storage subsystem to ensure that your information is the correct data every time. EMC's Symmetrix subsystem is the only storage system that guarantees that your data is both available and accurate.

Symmetrix offers a unique end-to-end integrated data verification technique consisting of codes and embedded ID on 4K clusters of data. Data integrity is verified from the host channel interface to cache to disk and back again, with the same data verification codes that are generated once at the entry point.

Additional Symmetrix Features

Multi-System Imaging

Symmetrix supports multiple System/390 environments through use of its 3990-3 or 3990-6 emulation modes and Hyper-Volume Extension feature. For control unit definitions of more than 64 device addresses, it is necessary to define multiple system IDs (SSIDs) with each SSID having a maximum of 64 devices. Symmetrix systems support up to 16 SSIDs with up to 64 devices per SSID up to a maximum of 1,024 logical devices per Symmetrix 5000 system. With IBM and PCM equivalents, up to eight-path connectivity may exist to any single device within the Symmetrix configuration.

Sequential Data Striping

Symmetrix family systems are fully compatible with IBM's Sequential Data Striping function for 3990 Model 3 and 3990-6 with Extended Platform in the ESCON environment. Sequential Data Striping automatically distributes accesses to balance the workload across disks. It also provides fast execution on large I/O-bound sequential processing requests by allowing I/O operations to be managed in parallel across as many as 16 devices. The Symmetrix system handles the smaller blocks of data provided by Sequential Data Striping by performing up to 16 concurrent I/Os over multiple paths.

Sequential Data Striping is available only with DFSMS/MVS (Data Facility Storage Management Subsystem) with storage management active. Symmetrix must be emulating 3990 Model 3 or Model 6 and running the appropriate level of microcode. It must be attached via ESCON channels and have SMS-managed volumes.

Host Data Compression

Host Data Compression compatibility is provided on Symmetrix 5000 systems via implementation of Sequential Data Striping support. The MVS instruction-driven data compression function is supported on high-end air-cooled and water-cooled IBM ES/9000 model 511/711 processors.

Multi-Path Lock Facility/ Concurrent Access

Symmetrix systems support the Multi-Path Lock Facility/Concurrent Access (MPLF/CA) for use with the ultra-high performance Airline Control Program (ACP) and Transaction Processing Facility (TPF) host operating system environments. MPLF/CA allows multiple concurrent I/O requests to the same logical device from multiple TPF mainframes. The Symmetrix system maintains the names and status of logical locks currently in use and responds to requests to obtain or release a lock. This allows multiple hosts to share DASD through multiple paths in an active OLTP environment while maintaining data integrity. MPLF/CA is an enhancement and replacement for the Extended Limited Lock Facility (ELLF) and Limited Lock Facility (LLF).

Partitioned Data Set Search (PDS) Assist

Symmetrix systems support IBM's Partitioned Data Set (PDS) Search Assist feature for 3990 Model 3 or Model 6 with Extended Platform in both ESCON and parallel channel environments. PDS Assist improves performance on large, heavily-used partitioned data sets by modifying the directory search process.

Chapter 3
EMC Enterprise Storage Networks (ESN)

Overview

EMC Enterprise Storage Networks (ESNs) are dedicated networks that connect multiple enterprise storage systems to all types of servers and their associated operating systems and applications. With a fast, reliable infrastructure for the common management, protection, and sharing of data across the enterprise, they relieve network bottlenecks and offer considerable cost savings. In this way, they enable cost-effective, high-capacity, heterogeneous information consolidation.

ESNs use Fibre Channel technology to provide EMC Symmetrix storage systems with connectivity, distance accommodation, and throughput rates that are significantly superior to those provided by traditional point-to-point connectivity. Fibre Channel is a highly reliable message transport protocol that provides the fastest, most scalable performance available today for connecting multivendor host servers and storage systems, either centrally located or dispersed throughout the enterprise. With an EMC ESN, organizations can consolidate hundreds of servers into a virtual data center spanning the entire enterprise. EMC's advanced software functionality for information sharing, protection, and management enables central monitoring and control of this virtual data center.

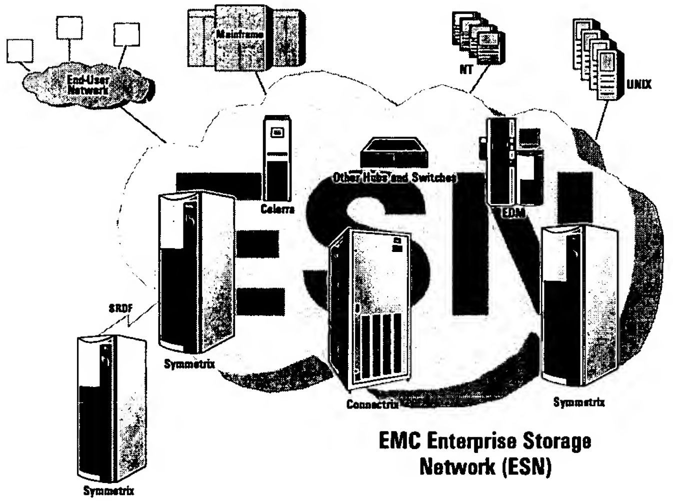
ESNs offer several major benefits:

Distance — ESNs enable customers to physically attach heterogeneous UNIX and Windows NT servers and EMC Symmetrix Enterprise Storage systems across great distances. Current implementations of SCSI impose a distance limitation of just 25 meters between servers and storage systems.

Consolidation — ESN connectivity enables organizations to consolidate data among widely dispersed servers and storage systems, while supporting the ever-increasing amounts and types of corporate data. Servers that were once outside the immediate vicinity of the data center now can leverage ESN technology to take advantage of the advanced information sharing, protection and management benefits of centralized EMC Enterprise Storage resources.

Enterprise Connectivity — Consolidation of multiple server types with multiple connectivity options requires maximum flexibility from the storage system. The ability of a single Fibre Channel-based enterprise storage platform to simultaneously handle not only mainframe parallel and serial channel connections, but also major open systems servers with FWD SCSI, Ultra SCSI, and Fibre Channel, enables a plethora of connections among mainframe hosts and open systems.

Channel Throughput — Until now, storage connectivity has been restricted to point-to-point, relatively slow connection schemes. By extending channel bandwidth, ESN enables customers to exploit the performance and functionality of EMC Enterprise Storage systems for key applications such as OLTP, data warehousing and Internet commerce.



The EMC Enterprise Storage Network™ architecture enhances the ability to consolidate, share, protect, and manage vital information from across the entire enterprise and gain maximum business value from it.

Chapter 4
EMC Enterprise Storage Solutions

Information Protection - EMC provides software solutions that maintain continuous data availability. The standard features of Symmetrix software solutions facilitate continuous data availability in the event of any major system component failure or power outage, and provide the ability to repair or replace the failed component without any interruption in operation. EMC Enterprise Storage software solutions continually perform self-diagnosis to identify and correct potential problems prior to any disruption of data availability. These software products include:

- Symmetrix Remote Data Facility (SRDF)
- EMC TimeFinder
- Symmetrix Data Migration Services (SDMS)

Information Sharing - EMC offers centralized, sharable information storage for supporting changing environments and mission-critical applications. This leading-edge technology begins with physical devices shared between heterogeneous operating environments and extends to specialized software that enhances sharing information between disparate platforms. These software solutions include:

- Symmetrix Enterprise Storage Platform (ESP)
- EMC InfoMover
- DataReach

Information Management - EMC consolidates storage capacity for multiple hosts and servers and improves information management. The Symmetrix Manager family of products further enhances this efficient, consolidated storage approach. These optional software solutions provide powerful GUI-based tools that simplify Symmetrix configuration, performance, and status information gathering and management. These software solutions include:

- Symmetrix Manager
- EMC PowerPath
- EMC Volume Logix
- EMC Data Manager (EDM)
- FDR Family of Backup/Restore Solutions

For more information about EMC Enterprise Storage solutions, contact your EMC sales representative.

Chapter 5
Services and Support

Professional Services

EMC Professional Services consultants provide a full range of services to enable you to extract maximum value from your information. These services assist you in applying EMC Enterprise Storage concepts and capabilities to your business issues. The EMC approach enables you to put information at the center of your IT infrastructure so you can take control of your information and utilize it to your full advantage.

Professional Services help you leverage EMC Enterprise Storage solutions, expertise, and resources to achieve success faster, more cost effectively and with less risk. They enable you to:

- Understand your current IT environment and take charge of it
- Create a more responsive, efficient, and flexible IT infrastructure with information at its center
- Share, protect, and manage critical information across the enterprise
- Deploy robust new enterprise solutions faster.

EMC Professional Services personnel utilize EMC Storage Logic™, a framework of EMC-specific and storage industry best practices that addresses all phases of an enterprise solution. Use of this framework ensures consistency and quality of deliverables and facilitates effective management of project budgets, schedules, and specifications.

To help you build an IT infrastructure that takes full advantage of all your critical information, EMC Professional Services provides both strategic enterprise consulting services and practical enterprise software implementation services. Consulting services help you assess your current infrastructure in light of your requirements and sort through various options. Implementation services help you integrate a specific hardware and software solution into your unique environment.

Enterprise Business Continuity

Enterprise Business Continuity services protect and enhance your ability to generate revenue. They help you build an enterprise business continuity infrastructure that not only eliminates unacceptable downtime (planned and unplanned) but also creates new ways to capitalize on business opportunities to generate increased revenue and customer services.

Enterprise Business Continuity services help you map and build your infrastructure to satisfy a range of business continuity requirements from high availability to mission-critical availability to continuous availability to disaster recovery. Assessment, planning and design, and software implementation assistance is available.

Use of EMC Professional Services personnel for implementation enables you to quickly realize the advanced functionality of EMC software, while your in-house IT staff continues with other revenue-generating activities. A range of SRDF software implementation services are available, from basic software installation to complex integration projects that encompass the complete project lifecycle. Regardless of the level of complexity, EMC Professional Services personnel can address your unique technical, staffing, or timing requirements.

In addition to software implementation services for such key business continuity products as EDM, SRDF, and TimeFinder, implementation services are also available to help you expand your information sharing, management, and protection capabilities by adding other EMC software products to your infrastructure.

EMC Customer Support

The EMC Customer Support Center, headquartered in the United States, directly supports EMC hardware and software products. The following numbers offer technical support:

U.S.:	(800) 782-4362 (SVC-4EMC)
Canada:	(800) 543-4782 (543-4SVC)
Worldwide:	(508) 497-7901 (or contact the nearest EMC office)

AS 1

EMC^2
The Enterprise Storage Company

EMC Corporation
Hopkinton
Massachusetts
01748-9103

1-508-435-1000

In North America
1-800-424-3622 ext. 362

www.EMC.com

RAB™
LISTED
http://www.raid-advisory.com/emc.html

NQA
ISO 9001
REGISTERED COMPANY

NATIONAL ACCREDITATION OF CERTIFICATION BODIES

EMC², EMC, MOSAIC:2000, and Symmetrix are registered trademarks and EMC Enterprise Storage, The Enterprise Storage Company, DataReach, EMC Storage Logic, InfoMover, PowerPath, SRDF, SDMS, EMC Enterprise Storage Network, and TimeFinder are trademarks of EMC Corporation. Other trademarks are the property of their respective owners. *RAB* is a Certification Mark of the *RAID* Advisory Board, St. Peter, MN, 507-931-0967.

©1999 EMC Corporation. All rights reserved.
Printed in the USA. 2/99

Product Description Guide
L702.4

factiva.
Dow Jones & Reuters

Features
New Products Make Replication Easier: An Analysis
Bruce Robertson
2,260 words
1 September 1993
Network Computing
99
Issue: 409
English
(Copyright 1993 CMP Publications, Inc. All rights reserved.)

Replication used to be a matter for third-party vendors to handle. You created some routines, often with the help of some company's data extraction utility, that ran every evening, perhaps downloading data from the mainframe and putting it out on a server in a format usable by decision support analysts running spreadsheets.This worked fine, except that you were the integrator. In some cases, you had to write code, or at least extensive triggers and stored procedures, inside your database management system.

Today, database vendors see replication as an important part of database management. They are designing products that help with this task, automating all or part of the replication process.

With reduced budgets in most shops for programming, buying instead of building a replication system can provide great economic benefit. Getting replication software from a single source also avoids finger-pointing between the utility and database vendors if you need technical support.

On the other hand, using someone else's software means it won't be exactly what you would have written yourself. If you are using Sybase or Oracle Corp.'s replication software, your databases will replicate their way.

To help you decide whether their way fits your needs, we offer here discussions of the automatic replication features available with two database products: Oracle version 7 snapshots and Sybase System 10 Replication Server. We look at how they actually work, and what they can and can't do. An accompanying article discusses Lotus Development Corp.'s Notes, its form of replication and how it differs essentially from replication for relational databases.

In addition, the case studies profiled on page 108 show other products in action, including Trinzic Corp.'s InfoPump and Red Brick Systems' Red Brick Warehouse. Each product offers a particular flavor of replication, but each has its strengths and weaknesses.

Oracle 7 Snapshots

Oracle's new version 7 relational database management system (RDBMS) supports a simple but effective automatic replication feature called the snapshot. The database administrator can specify that a given database server should copy data from an original, or master, table, creating a read-only snapshot, or copy. Any number of snapshots can be set up on different database servers, all copying a single master table.

There are simple and complex snapshots. Simple snapshots are replicated from a single table to a similar table on a separate node. Snapshots can include a subset of columns or rows from the original master table, as declared when the snapshot is created with SELECT and WHERE clauses. Snapshots are created in a simple SQL declaration, including a directive as to how often the snapshot should be refreshed. The system handles everything from that point on. There is no extra programming of stored procedures or triggers; Oracle creates appropriate triggers automatically when the snapshot is declared.

Simple snapshots provide options for snapshot updates that are not full refreshes (complete copies) of the original table. The database administrator can create a snapshot log on the master table and then snapshots can update only the transaction changes by coordinating with the snapshot log. This changes-only update typically involves much less data and therefore much less traffic on the intervening networks, as well as less load on the database server nodes involved. Multiple snapshots can be supported from a single database table using the snapshot log as well.

Complex snapshots replicate more than one table at a time to a single remote table, as defined with joins in the WHERE clause of the snapshot declaration. Complex snapshots cannot benefit from a fast, changes-only refresh; they can be updated only with complete refreshes.

The alternative to a complex snapshot is multiple simple snapshots, with a new view locally that joins them. Although this may be more efficient, since each table benefits from the fast refresh, a loss of data integrity is a possibility. There is no guarantee that the two tables being replicated separately are themselves in sync when the replication copies them sequentially. The second master table could have been changed while the first was being refreshed to the snapshot. Then, when the second table was replicated, the two resulting snapshots might no longer be consistent.

Given these risks to data integrity, when would snapshots be helpful? They are effective if the data to be replicated changes very infrequently or is confined to a single table. For relatively static data, a snapshot is easy to set up. Of course, applications have to be written to read from the snapshot instead of the master table, and if it is necessary to write to the database, the writes will have to update the master table since the snapshot is read-only. Writing to the master means, however, that the snapshot would not be updated until the next refresh, which could be hours away or even overnight.

Since snapshots aren't replicated in real-time, they can be expected to be only loosely consistent with the master table. Still, most organizations have data that is relatively static or that should be read-only in most sites, so the snapshot feature will help with application development and deployment on complex networks.

Other difficulties with managing the ongoing snapshot replication environment can arise, however. If a network link is down and a scheduled snapshot is missed, the good news is that the next snapshot refresh will include all the required changes. The difficulty will be in keeping track of when successful refreshes have occurred, particularly if there are a large number of snapshots of a single master table in multiple geographic locations. If one site is unavailable to do a snapshot, it will be out of sync with the other remote sites until it comes back on-line and is refreshed.

Such a one-to-many setup could become inconsistent over all sites if the master table is being updated while a large number of different site refreshes are under way. There are no specific snapshot management programs built-in to give the database administrator a quick status report on all refreshes.

Oracle touts another replication approach for Oracle 7: a synchronous replication solution that requires custom programming. Triggers can be set up to write data to other database servers. This works neither automatically nor even transparently, and in fact is no different than what any database system supporting stored procedures could do.

Sybase Replication Server

Sybase, with its new System 10, will provide a Replication Server option. Due to be available on a variety of Sybase Unix platforms by the end of 1993, this option is expected to provide for replication services appropriate to a variety of applications, and it can be implemented in many systems with little extra programming. Sybase also plans to ship applications to manage the replication service and keep database administrators informed of the current status of replications.

With Replication Server, the database administrator can specify tables, sets of tables or subsets thereof to be replicated from a primary server to any number of secondary or mirrored servers.

To set up data replication, the database administrator writes a replication definition for each primary table, listing its columns and data types. This makes the data available to be replicated. Then, on the remote systems, the administrator creates empty tables with exactly the same columns and data types as the primary tables. Finally, the administrator creates a "subscription" for the rows of the tables available for replication at each remote site. After that, any change to the master table will be replicated to the copies in the terms defined by the subscription relationships.

The replication mechanism handles complete transactions, which may update several master tables, making sure all the updates are replicated to the other systems. As a client application writes to the original database server tables, a special Log Transfer Manager process notes the update. The Replication Server then causes those changes to be executed for each subscribing copy.

If the copy's table is large, Sybase supports a first-time initialization from tape to avoid having to copy the entire table over the network. From initialization on, only changes to the copy are sent over the network.

All the copies are read-only, so updates to the database must be made to the primary table. Sybase, however, has an asynchronous stored procedure feature that can apply a loose-consistency approach to updates from remote sites as well.

The Replication Server requires remote updates to be initiated by server-based stored procedures, which are called like any Remote Procedure Call (RPC) by user applications. If all transactions are run via these RPCs, the Replication Server can redirect the RPCs and apply them to the master tables directly, instead of to the local tables that are replicas.

Since the update is queued up by the local Replication Server, the master table doesn't have to be directly available for the user to have made a guaranteed (though possibly delayed) transaction to the database. When the connection is restored, the local Replication Server updates the master table, whose change, in turn, triggers an update of the local table.

The asynchronous nature of the updates makes the Sybase approach, both for replicating and remote updating, resilient to intervening WAN outages or other failures. If there is not real-time access to the other sites, transactions are stored up to be migrated when the link comes back up. A two-phase commit mechanism does not protect the entire process; rather the Replication Server ensures integrity in its own mechanism.

There are drawbacks to this asynchronous approach. First, there is no protection against having different replica sites out of sync until the link is restored and updating can take place. Second, any user doing an update from a remote site has to wait for the RPC to be executed at the master, and then for the Replication Server to replicate that update back to the read-only copy local to the user. This delay in changes being reflected in the local database may be unacceptable for certain applications. Even if real-time links are up and bandwidth is high, there will be some intermittent time when the data is inconsistent.

Finally, if a transaction is delayed, new business problems may need to be resolved. If, for example, the delayed transaction involves deleting a given product from a central inventory, another site needing that part might take them all before the delayed transaction can be completed. There are obviously business needs better served by a synchronous two-phase commit replication transaction or a direct master table transaction than by asynchronous replication.

Sybase's architecture can help solve database problems that require near-real-time performance without a delay when a WAN link fails, however. For example, several branch sales offices could have write access to account records for their respective territories on their local servers, and all the offices' information could be replicated to a central site and back out to the branches, so they each have data from all the rest. The central site might be responsible for decision support applications. If the WAN links to the central site were down, or the central system were slowed down by large report runs, the remote sites still would be able to query and update their accounts' status locally and have a relatively recent picture of the whole organization. Changes would be migrated up to the centralized copy later.

Sybase also has built in some intelligent routing capability. Updates for European sites, for example, may all physically go from New York to London on one WAN link, and then via separate links to other European sites. The Replication Server can send a single transaction from New York to London, and then let London relay separate transactions to each European remote site.

Replication Isn't Simple

None of the replication solutions we've profiled solves all problems. Still, any one could be the tool that unlocks a particular application required by your organization a function you couldn't have implemented before as easily or at all.

The most obvious weakness of the three solutions we profile, compared with writing your own code or using third-party utilities, is that these solutions require the same vendor's database product on all servers involved. While it is actually possible for the Sybase and Oracle products to update other vendors' products, this functionality requires the use and expense of vendor-specific gateways to those data sources. For such multivendor data replication needs, products such as Trinzic's InfoPump can be better solutions.

The second obvious weakness of most replication systems is that they replicate data to read-only copies.

But if users in a remote site also need to make simultaneous updates of the data in the replicated tables, then replication becomes much more problematic. Generally, updates are handled without replication directly to the master table, requiring real-time access to that master system. This leads to a decentralized-read, centralized-write data architecture.

Only two of the products we profile offer some level of asynchronous remote update without requiring direct access to the master table at transaction commit time. Sybase's Replication Server supports asynchronous remote procedures. Lotus Notes allows for remote updates by restricting the data to different versions, essentially allowing only CREATES and DELETES, but not updates.

Getting all the details of replication straight and maintaining data integrity are complex tasks, and trusting a respected vendor to provide automated replication solutions can simplify your life immensely.

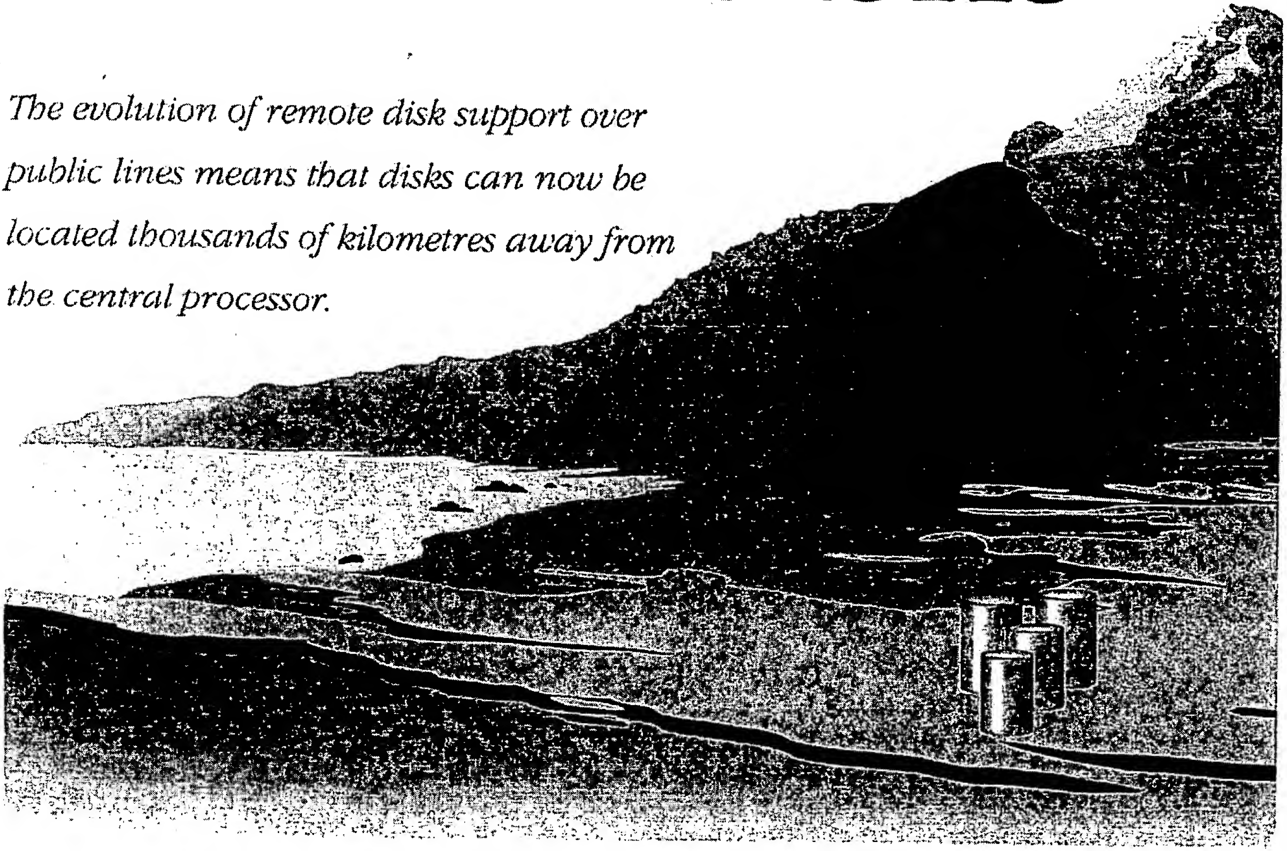
Document nwcp000020011101dp91000Su

© 2005 Dow Jones Reuters Business Interactive LLC (trading as Factiva). All rights reserved.

AT1

Remote disks

The evolution of remote disk support over public lines means that disks can now be located thousands of kilometres away from the central processor.



When IBM announced its ESCON (enterprise system connectivity) fibre optic networking scheme in 1990 it opened up an intriguing new possibility for its mainframe users: they could put their processors in one place, and their disks in another. And those two locations could be up to nine kilometres apart.

This aspect of the ESCON technology has clear business implications: companies requiring easily accessible off-site back-up can write copies of their data, in real time, to disks situated away from their main processor. And those organisations short of space at their main offices can write to disks several streets away, or across a river in a less expensive part of town.

But ESCON has several limitations when it comes to remote communications: the maximum distance of 9 kilometres between the processor and the disks (27 kilometres with repeaters) is too close for many disaster recovery requirements, especially in the US, where there is an ever present danger of earthquakes in many areas. And the nine kilometres is too close for most companies to take advantage of lower rents at more remote locations.

But equally important, prospective ESCON customers are not usually allowed to dig up the roads to lay their cables. Although a few customers have very large sites which allow large fibre networks to be laid (BA at Heathrow Airport, for example), most mainframe customers will have to restrict their use of ESCON to local sites (for example, for linking devices between the floors in large buildings).

These limitations, along with the cost, have restricted the appeal of ESCON, even though it significantly increases the speed at which IBM mainframes and peripherals communicate (to 17 megabytes per second) and enables resource sharing of disks at the local site. The interest of many customers, however, has been stirred by IBM's endorsement of the use of optic fibre channels for distributing high speed peripherals.

CHANNEL ALTERNATIVE

Until recently, there was no way to distribute disks across a wide area. The only way to get over the ESCON distance limit has been to use lines and circuits supplied by the PTTs (Postes, Telegraphes and Telephones - a term for the big national telecommunications suppliers) such as Mercury and British Telecom. But while these lines have long been used to distribute slower peripherals such as printers and tape drives, writing to disks requires both speed and careful synchronisation which public lines have not been able to support.

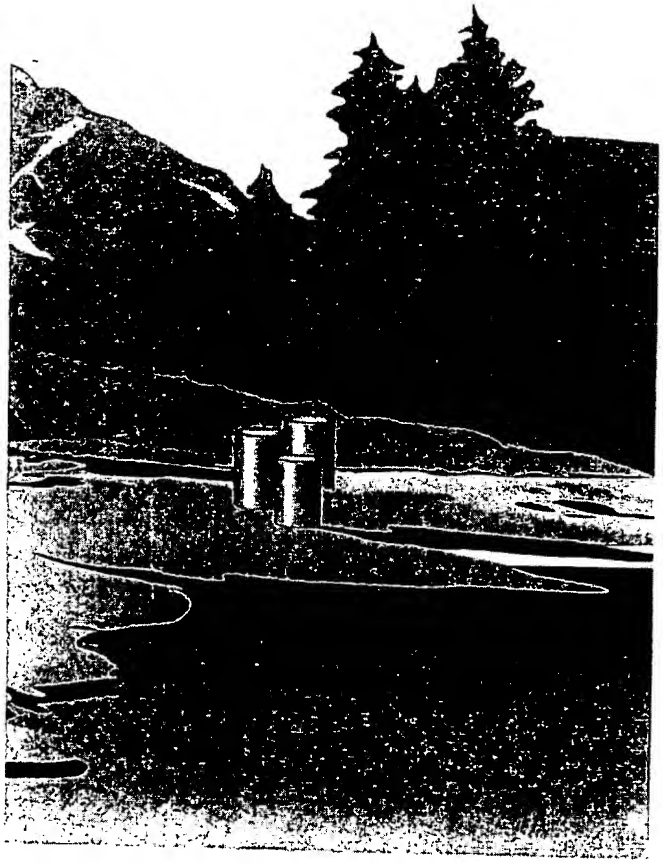
It was these limitations which led Jeffrey Davis, planning manager for AT & T Paradyne, a leading supplier of channel extenders, to say that '..direct access storage devices (DASD) will not operate over remote, high speed data communications lines. No-one has a global solution to DASD'.

However, Network Systems Corp (NSC) and Computer Network Technology International (CNTI) both say that they have solved the problems of supporting remote disks over public lines using their standard channel extension equipment. As a result, both say that users can put extra or back-up disks wherever they want - even thousands of kilometres away.

The only condition is that the high speed lines (conforming to the international E3 or T3 standards, which run at between 34 to 44 megabits per second) are available from the PTTs.

The use of channel extenders to support remote disks opens up new possibilities. A bank of remote disks, for example, can be used for back-up, for providing data disaster recovery support, and for providing extra disk storage during peak periods; and, by using dual-copy software techniques

Supplied by the British Library - "The world's knowledge" www.bl.uk



(now available with both the IMS and DB2 databases), remote back-up can now be instantaneous.

This can give large cost savings. Most customers requiring immediate back-up install a second processor, complete with operators, communications equipment, disks, and software licences and then link the sites together using T2 or T3 lines. Effectively, the user is forced to duplicate all the costs of running a data centre.

There are other applications. General Accident, the insurance company, used NSC's Remote Data System to write to a second set of disks while preparing to move data centres. And an international financial institution is planning to use the technology to ensure that two sets of financial trading information are exactly mirrored at its London and New York offices.

HOW IT'S DONE

Channel extenders make the mainframe software believe that a remotely situated device, such as a tape drive or a printer, is actually locally attached.

A channel extender has buffer memory, stores data temporarily, passes on and filters out certain acknowledgements, and performs the address filtering to transfer only the traffic destined for the remote device. This is important, as the IBM channel may be operating at 2 megabytes, 4.5 megabytes or higher while the communications line may only be running at between two and 44 megabits. A channel extender is required at each end of each communications line.

The use of channel extenders for sending data out to tapes and printers is now routine, but disks present a different problem. The host requires careful synchronisation with the disk controller, and will not proceed until it has received confirmation of each disk write. This extends the response times and can cause MVS to conclude that there has been a disk write failure.

There are three approaches to this problem, each with its own problems. The first approach is channel spoofing, where the channel extender tells the host that the write has been completed. The extender is then responsible for holding the data temporarily and ensuring that it does get written to disk. If there is a disk failure, it must then tell MVS that there has been a disk write failure after all.

This approach, taken by CNTI, is fast, efficient and involves no host software; it has proved particularly successful for supporting tapes and printers. Other suppliers such as AT&T Paradyne and Data Switch also use this technique.

But NSC claims it carries risks because MVS is being told data has been safely stored when it has not. 'If there is a failure, it has to go back to the host. You are taking error correction away from MVS,' says John Cunningham of NSC.

A second technique is 'signal racing'. This effectively means that no differentiation is made between local and remote devices. This is mainly used for local extensions (a few hundred metres) and is considered impractical for disks and longer distances. It is best suited for solid state devices because they are faster.

NSC uses a third approach: it puts software onto the host which deals with the delays and informs MVS of what is going on. The software is also able to add extra error correction, performance enhancement processes and alternate path routing.

The drawback with this approach, says Peter Dixon, managing director of CNTI, based in the UK, is that it is more complex, more expensive and uses up processor resources. Indeed, he claims that during a period of very high activity, up to eight per cent of processor resources could be used. If the computer is already heavily loaded, potential cost savings could be wiped out.

In practice, both CNTI's and NSC's approaches are likely to satisfy most user's requirements. One clearing bank has been studying both techniques (although it has not begun trials) and believes both could meet its requirements.

BUSINESS CASE

So far, only a handful of customers worldwide use channel extension technology for the support of remote disks, and many of the leading vendors (McData, AT & T Paradyne, Data Switch) have stayed out of the market.

NSC and CNTI report high interest in their devices, but also acknowledge that the use of remote DASD is only likely to be useful in certain situations. Those seeking space savings, for example, might be better off installing small footprint disk arrays (says CNTI), while those seeking very high volume disk back-up will probably need to install a remote host (says NSC).

Those most attracted to the use of remote DASD will be those who want to make the best use of existing disk resources at different locations, or want some form of real-time back up.

□ *Andrew Lawrence*

USER EXPERIENCE: CAISSE POPULAIRE (MONTREAL)

The use of channel extenders to support disks at remote sites is only just out of the pilot phase. Caisse Populaire, a Canadian financial institution, is believed to be the first major company to use the technology.

Caisse has a large IBM network based around an IBM ES/9000 Model 820 mainframe. It has over 10,000 terminals and 1,450 teller machines in the network. At its peak, the bank's Montreal headquarters deals with over 300 transactions per second.

Caisse began looking at remotely attached disks as part of a review into disaster recovery and security procedures. Caisse's normal procedure was to journal each day's transactions onto disk and then archive them to tape; in addition, any changes were recorded in a transaction database and these were transmitted every 15 minutes to a back-up site 15 kilometres away.

This procedure had a flaw: if a disaster occurred, Caisse was concerned that it could lose up to 30 minutes worth of transactions, totalling 300,000.

The solution lay in real time journaling. Using a high speed T3 telecommunications line running at up to 44.7 megabits per second, Caisse linked a bank of Hitachi Data Systems disks at the back-up site to its mainframe via two Network Systems Corp remote data system (RDS) channel extenders. NSC software is also run on the host.

Using this equipment, a secondary log of the IMS database banking application is written in real time to the remote disks. No second mainframe (or the software licences involved) is necessary. Tests show that the disk response time is constantly maintained at about 17 milliseconds, faster than the bank had initially expected. It is also possible to write to both tape and disk in about 28 milliseconds. The function of MVS and IMS logging is unchanged.

Caisse Populaire believes that it will save about $100,000 per year as a result of the direct remote disk support.

Supplied by the British Library - "The world's knowledge" www.bl.uk

AR2

PROTECTING Your Data

By Steve Bobrowski

OVERVIEW AND COMPARISON OF BACKUP AND RECOVERY FEATURES IN DATABASE SERVERS.

Steve Bobrowski is the president of Animated Learning, a San Francisco Bay Area firm that offers consulting, educational services, and educational software for relational database management systems. You can reach him at Internet address stevebob@netcom.com.

As Murphy's Law states, accidents and problems are an inevitable consequence of life. Unfortunately, there are many problems that can rear their ugly heads in the database management system world. Abnormal shutdowns such as a power failure, user errors such as an accidental disk format, or unexpected storage device failures such as a disk crash are all examples of situations that can cause loss of data in a database system. Therefore, when considering database servers, it's important to understand the features that protect the valuable work, data, and availability in a database system.

This article explains some of the common components that database servers use to protect databases from problems. Then we'll see how five different database servers compare in the area of backup and recovery features.

Protective Features

A database server usually has a number of features that it uses to protect a database from problems. The sections following explain some of the most common features seen in today's database servers.

The Transaction Log. The flight log in the cockpit of a commercial airliner records what goes on during a flight. If the plane crashes, investigators can use the flight log to reconstruct what went wrong.

Similar to the flight log in an airplane, database servers keep a log of the changes made by transactions. If a database experiences a problem, the server can use the database's transaction log to reconstruct changes made to the database by committed transactions.

Log Structure. In general, all database servers have a similar transaction log mechanism. The server fills the log as transactions change the database. At the same time, the server archives older log entries to backup storage (tape, for example), thereby freeing space in the log for new entries. Using this scheme, the server creates an archived transaction log that is a permanent, growing history of changes to the database, and uses a controlled amount of space to perform real-time transaction logging.

The structure of a server's transaction log can have a significant effect on performance and application limitations. Some servers, for example, employ a single file for the transaction log. With a single file, there is an inherent contention between writes and reads on the same disk, as the server tries both to log new transactions and archive old ones. In contrast, multifile logs can alleviate this problem, because you can place the different files of the log on different disks.

Automatic Log Archiving. Some servers offer the capability to archive log entries to permanent backup storage automatically. As noted above, archiving older log entries frees space in the log for new entries. If a server doesn't offer automatic log archiving, the administrator has to monitor the transaction log manually. As the log fills, the administrator manually issues a command to archive the log to ensure that it will have adequate space for ongoing transaction logging.

File Backups. To recover a database from physical damage, you need backups of the files that comprise the database. When someone accidentally formats a disk that stores some of a database's files, for

Supplied by the British Library - "The world's knowledge" www.bl.uk

example, you have to restore backup copies of the lost files before you can recover the database using the transaction log.

Online Backups. Critical applications require high availability and cannot afford frequent periods of down time for regular database backups. Therefore, most database servers support online database backups — the option to back up a database while it is open and in use. Later on you'll see how different implementations of an online backup mechanism can affect application performance during online backups.

Incremental and Complete Database Backups. When a server performs a complete database backup, it copies all pages in the database to the backup. To reduce the amount of time necessary to back up a database and the impact it may have on concurrent application performance, some servers also offer the capability of performing incremental backups. When a server performs an incremental backup, it copies only the pages that transactions have changed since the last complete or incremental backup. During the restore process, the server first restores the most recent complete backup. Then, the server updates the restored database using any successive incremental backups. Finally, the server applies the changes in the transaction log to recover the database.

File Mirroring. Most database servers allow you to mirror some or all of the files that make up a database. When a server mirrors a file, it writes the file in parallel to two or more different locations. If a disk failure damages one copy of the file, the other copy remains available so that the database system can continue to operate without interruption and without any requirement for recovery.

Understand that the transaction log is the critical component for a server to mirror. Even if you lose all data files, you can always recover a database using a data file backup and an intact log. However, if a disk failure damages the log, you can't apply the log to achieve complete recovery. Data file mirroring simply provides added protection if your applications require high availability.

TABLE 1

	Gupta SQLBase 5.1	Informix Online 5.01	Ingres 6.4	Oracle7	Sybase SQL Server 4.9
Transaction log structure	Multiple files on demand	Multiple files	Single, static file	Multiple files	Single, dynamic file
Log archiving	Manual	Automatic	Automatic	Automatic	Manual
Type of online backup	Consistent	Consistent	Fuzzy	Fuzzy	Consistent
Incremental backups	No	Yes	No	No	No
File mirroring	No	Yes, all files	No	Log files, but not data files	Yes, all files
Group commits	Yes	Yes	Yes	Yes	Yes
[illegible]	Yes	Yes	Yes	Yes	Yes
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

A comparison of the protection components of five different database servers.

Performance Considerations. *Group Commits.* Transaction logging involves disk I/O, a very expensive operation that can slow system performance. Therefore, database servers have all sorts of tricks to minimize disk I/O and get the best performance.

One such trick is a group commit. When you commit a transaction, the transaction is not "committed" until the server writes all of the transaction's changes and submits a final commit record to the log. A commit record indicates that a transaction's changes are committed in the log. To further reduce disk I/O bottlenecks when logging transactions in a heavily loaded system, some database servers perform group commits. When multiple transactions barrage the server with commit requests at or near the same time, the server can group multiple commit record writes into a single I/O operation, thereby reducing disk I/O and increasing system performance with respect to committing transactions.

To find servers that perform in logging and other areas, look for servers that use other I/O minimization techniques such as multipage and asynchronous read/write I/Os.

Consistent vs. Inconsistent Online Backups. If online backups are something you need for your applications, you should take a close look at the implementation of a server's online backup mechanism to determine its effect on application performance.

An online backup can negatively affect the ongoing performance of a heavily loaded system if the server has to make a consistent backup of a database. A consistent backup means that every page in the backup of a database is consistent with respect to the same point in time. To make a consistent backup while a database is in use, the server usually performs several operations.

The server first writes all modified pages in memory to disk so that all pages in the database are consistent. Most systems call this a checkpoint or a consistency point. Then, the server begins backing up the database, page by page. When a transaction tries to update a page already written to the backup, the server allows the transaction to proceed and update the page. When a transaction tries to update a page not yet written to the backup, however, the transaction must wait until the server writes the requested page to the backup. Most servers work to reduce an online backup's negative performance effect on applications by coordinating consistent page backups with ongoing transaction requests.

In contrast, an online backup does not detract from the performance of applications if the server makes inconsistent or fuzzy online backups. A fuzzy backup is

Supplied by the British Library - "The world's knowledge" www.bl.uk

one where the pages of a database are not necessarily consistent with respect to a single point in time. The server simply backs up a database page whenever it gets to it. At the same time, the server allows a transaction to update a page not yet written to the backup because there is no requirement to make the pages of the backup consistent.

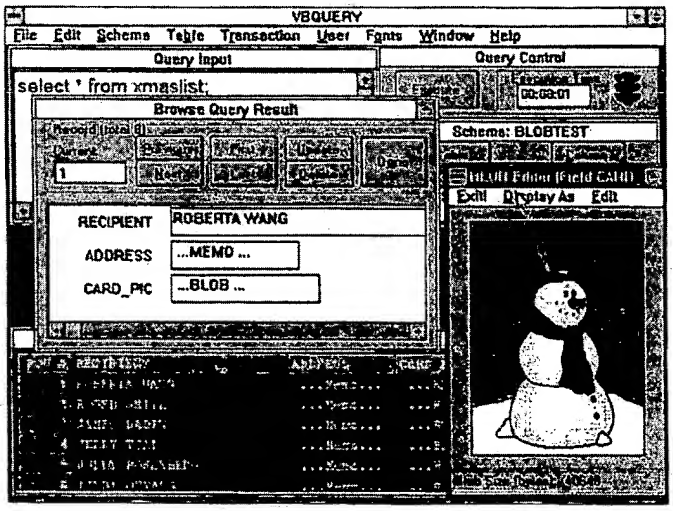
At first, you might be concerned that fuzzy backups can jeopardize the consistency of a database. However, remember that you use a backup only for database recovery. During recovery, the server applies the transaction log to reconstruct transaction changes in the pages of the database. Therefore, after the server applies the log during recovery, the database is left in a consistent state.

Recovery

Protective features of a server are great, but a database server also needs recovery features so that you can actually recover a database when necessary. The next few sections explain some of the common recovery features found in many database servers.

Fast, Full-Featured SQL Engine for Windows

Ideal for mobile computing as an extension to Client/Server systems and applications that run on small to medium sized LAN file servers.



select * from xmaslist;

RECIPIENT	ROBERTA WANG
ADDRESS	...MEMO...
CARD_PIC	...BLOB...

Quadbase-SQL for Windows v2.0 is an industrial-strength SQL relational database engine, bundled with a set of powerful tools, that allows Windows developers to build applications using various languages like **Visual Basic, Realizer, Toolbook, SQLWindows, C, C++, ObjectView**, etc. A unique language-independent embedded SQL interface included with the system makes using any Windows language easy. The SQL engine, implemented as a DLL, is fully ANSI SQL 86 level 2 compliant, and supports the Microsoft ODBC standard, which provides **seamless, object-oriented** access from Visual Basic v2.0. Extensions include referential integrity, scroll cursors, and outer join, along with other advanced features such as multi-user concurrency control, crash recovery, transaction processing, multiple instances, **BLOB** data, and read-only schemas for CD ROMs. The engine is very fast and compact, and is especially designed to manage large amounts of data efficiently. Visual Basic development is enhanced by embedded SQL and custom controls for browsing and data entry. dQUERY, an award winning query tool/report writer, and VBQUERY, an interactive query tool for Windows, are included for quick prototyping of SQL statements. A 'C' language API is also provided together with an embedded SQL preprocessor. The native file format is dBASE. This product is ideal for small to medium sized LAN file servers, notebook, or pen-based systems. Users can benefit from advanced relational database features while opening a migration path to SQL servers.

Call for a free demo disk now. Find out why AT&T, Hewlett Packard, Nike, EPA, GE and many more major organizations use Quadbase products.

Quadbase Systems Inc.
790 Lucerne Dr. #51
Sunnyvale, CA 94086
Tel: (408) 738-6989
Fax: (408) 738-6980

CIRCLE READER SERVICE NUMBER 159

Recovery from Abnormal Shutdowns. Most servers automatically recover a database from an abnormal server shutdown such as a power failure. When you restart the server, it sees that the system experienced an abnormal shutdown and it automatically applies the transaction log to the data files to recover the database before reopening it.

Recovery from Disk Failures. When you encounter serious problems such as a disk crash, you need to perform several steps to recover a database. First, you need to restore the database from a backup. Some database servers require that you restore by using a complete backup of the database, which means that you restore both damaged and intact files of the database. Other servers require that you restore only the damaged files. The latter case obviously results in a quicker recovery, especially if your database is large and only a small portion is damaged. If your server allows you to restore different parts of a database at the same time (such as, in parallel), you can also reduce the time it takes to restore a damaged database.

After restoring the damaged database, you instruct the server to roll forward the database by applying the transaction log. After roll-forward recovery is complete, the database contains all of the work that was committed before the failure.

Online Recovery. Some database servers allow for the recovery of a damaged portion of a database while the remaining intact portions of the same database are online and in use. Online recovery permits applications to function if the necessary tables are not in the damaged portions of the database.

Recovery to a Point in Time. Many database servers can recover a database to a point in time that you specify. For example, assume that a program runs amok and mistakenly deletes many rows and then commits the transaction. To get back the lost rows, you need to recover the database to the point in time just before the errant program went crazy.

Point-in-time recovery usually involves several steps. First, you need to do a complete database restoration using a backup of the database that was taken before the desired end point of recovery. Then, you can start roll-forward recovery, stopping application of the transaction log at the last committed transaction just before the end point of desired recovery.

Product Reviews

Now that you have an overview of many backup and recovery features, Table 1 (page 56) and the next few sections compare five different database servers in backup and recovery features. The sections are in alphabetical order.

Supplied by the British Library - "The world's knowledge" www.bl.uk

Gupta SQLBase 5.1. Gupta SQLBase 5.1 uses a multifile, dynamic transaction log. SQLBase automatically creates a new log file when the current one is full. To free disk space, the administrator must monitor the system for filled log files, archive, and then delete them.

For high availability, SQLBase supports online backups. However, it makes consistent database backups and does not have an incremental backup option to expedite backup time.

SQLBase 5.1 does not support file mirroring of any database component. Therefore, if you lose any file of a database to a disk failure, the product eventually halts. Even worse, if you lose a database's log file to a disk failure, you cannot recover the database. Your only option is to restore the database from the most recent backup and sacrifice losing the committed work performed after the backup.

In the area of recovery, SQLBase 5.1 supports complete recovery of all committed work from a disk failure using backups and roll-forward recovery with the log. It also supports recovery to a point in time. You must perform all recovery procedures while the system is offline.

In conclusion, Gupta does not attempt to market SQLBase 5.1 as a leader in the area of backup and recovery, especially for large databases. The server focuses on delivering a number of key protection features that are easy to use and typically adequate for smaller, less demanding database systems.

Informix-Online 5.01. Informix-Online 5.01 has a multifile, static transaction log. It uses the static files in a cyclical fashion. After filling one file, Online continues logging to the next available file. After it archives a filled log file, it can reuse the file the next time the system needs a new log file. Online also has the option to automatically archive filled files in the log, directly to tape if you so choose.

One interesting fact is that Informix-Online's multifile log limits the size of a transaction. That's because the product also uses its log to store transaction undo information. Consequently, until a transaction commits or rolls back, Online cannot write over the undo that would be necessary to roll back the transaction. To avoid a "long transaction" from preventing log file reuse, you can calculate and set thresholds that tell the server when to roll back long transactions automatically.

As its name indicates, Informix-Online has several features for sites that require high availability. Online supports online backups, although this type of backup is consistent and can slow applications because of contention problems. You can, however, minimize backup time and contention periods using incremental backups. In fact, Informix-Online is the only server reviewed that currently offers an incremental backup option. It also supports file mirroring of every component in a database to protect database availability from isolated disk failures.

As strong as Informix is in the area of protection, it is somewhat weak in the area of recovery options. While Online can completely recover a database from an unmirrored damaged data file, its recovery process must restore the entire database serially through a single storage device. If you have a large database, the time to restore the database before even commencing recovery can be substantial.

Informix officials indicate that Online 6.0, the product's next release, has options to enhance backup and recovery. For example, Online 6.0 will include the option to back up and restore a database by individual dbspaces (as in, database partitions) in parallel and configure-automated backups.

In summary, Online 5.01's backup and recovery features are best-suited for medium- to small-size systems, especially those that need to ensure high availability. If you take advantage of Informix's mirroring capability to protect every file in a database, you'll likely never have any problems. If you have a large database, however, backup and recovery can take a long time.

Ingres 6.4. The Ingres 6.4 protection mechanisms are somewhat different and more complicated than those of the other servers in this article. Therefore, before I explain the guts of the different Ingres 6.4 backup and recovery features, I need to introduce them and explain how they work.

Unlike other servers, Ingres 6.4 uses two separate structures to record database changes — a transaction log and journals. An Ingres recovery includes restoring the database from the most recent backup and then applying the log to roll back all changes since the start point of the backup. After applying the log, you have a consistent snapshot of the database as it was at the beginning of the backup. You can stop here, but if you do, you lose all committed changes since the backup. To recover the committed changes (since the backup), you then apply journal files to roll forward the database. As you'll see in a moment, you have to use Ingres journaling correctly or else you can get yourself into hot water.

Ingres 6.4 employs a single-file, static transaction log. When a sufficient portion of the Ingres log file is full, Ingres automatically archives committed log entries to back up storage so that it can free space in the log file. The server can archive log entries directly to tape.

It's important to note that because Ingres uses a single, static log file, the size you choose for the log file can limit transaction sizes in your system. You can't, for example, log a large data load in one big transaction if it creates more log entries that can fit into the Ingres log file.

Ingres 6.4 can also journal tables to provide an option for complete recovery from a disk failure. During recovery, Ingres can roll forward changes to the tables that you journal. Ingres lets you pick and choose which tables to journal, however, and this can get you into trouble if you do not consider what you are doing. For example, you can end up with an inconsistent database after a recovery if you accidentally journal a child table but forget to journal the child's master table. (There might be children without a master.) To make it simple and safe, I suggest that you journal all tables in an Ingres database.

Ingres 6.4 is one of two servers reviewed that supports fuzzy online backups. Therefore, Ingres does not degrade application performance during an online database backup, even in heavily loaded systems. Ingres also offers an option to back up directly to tape.

A useful feature is Ingres' special configuration file that you can use to customize your system with several advanced backup and recovery options. For example, you can set configuration parameters so that Ingres backs up data files on different disks in parallel for faster backup times, especially for large databases.

Ingres 6.4 does not support file mirroring of any database component. If you want the fault tolerance of mirrored files, you must use hardware mirroring.

If you're interested in futures, Ingres officials note that the next release of Ingres will include a mirroring capability for the transaction log.

I've already explained how an Ingres complete recovery works. Also note that you can specify a point in time to stop the application of journal files during the roll-forward phase of an Ingres recovery. And although it is extremely tricky to use, Ingres also allows restoration of individual tables using operating system backup files of database tables. However, use this option with extreme caution to account for inconsistencies among related tables (parent-child table relationships, for example).

In conclusion, Ingres 6.4 has a number of key backup and recovery features, but, in general, they are complex to understand and use. Given the above and Ingres's lack of file mirroring, Ingres best fits midrange systems without stringent

(continued on page 80)

Supplied by the British Library - "The world's knowledge" www.bl.uk

In the area of distributed database, DDCS/2 version 2 still does not include support for Query Manager as a front end on a client workstation. Front-end support is limited to the Command Line Interface or custom application programs. DDCS/2 still does not support remote joins or two-phase commit, so although an application can access and update multiple remote databases, it must also assume responsibility for joining the data if needed.

■ Buyers are looking increasingly for capabilities in their database servers that are currently unavailable in DB2/2, including stored procedures, scrollable cursors, and online backups.

Physical database file-placement options have not changed significantly with DB2/2. You still cannot, for example, stripe database and transaction log files across multiple drives to increase performance. And the PC/IXF export file format is still distinct and incompatible with the mainframe IXF format. The manuals suggest using delimited ASCII files to export data from a DB2 or SQL/DS database, and to import it into a DB2/2 database.

What does It Cost?

IBM has changed pricing slightly from Database Manager, with single-user installations getting cheaper and multiuser servers getting more expensive. Single-user DB2/2 costs $425. Client/server DB2/2 costs $2495 per server plus $75 per client. Single-user DDCS/2 costs $500 and multiuser costs $4680.

Conclusions

Even though DB2/2 is a substantial improvement over Database Manager in terms of performance and DB2 compatibility, it is not a product that can compete with the leaders in the LAN database server marketplace on features alone. Buyers are looking increasingly for capabilities in their database servers that are currently unavailable in DB2/2, including stored procedures, scrollable cursors, and online backups.

On the positive side, DB2/2's enhanced compatibility with DB2 will undoubtedly give it a boost over the earlier Database Manager product. You can use it as a development platform for databases destined to be deployed on mainframes. Best of all, DB2/2's ability to participate in DRDA means that when you transparently move the database to more powerful hardware, if necessary, and without change, you can continue to run applications developed on PCs. Although DB2/2 offers improved DB2 compatibility, XDB Systems still claims superior compatibility for its XDB RDBMS. Bing Yao, president of XDB Systems, says, "In our view, DB2/2 is basically a performance upgrade, not a compatibility upgrade."

In summary, DB2/2 will have a stronger appeal than Database Manager to shops already using IBM's other relational DBMS products, but it will probably not convert many already running non-IBM LAN environments. The future of DB2/2 is heavily tied to OS/2 2.*x*, and as a database consultant I presently don't see too much demand for OS/2 2.*x*. In a way this is unfortunate because OS/2 is a worthwhile operating system. If the market share of OS/2 2.*x* grows, I suspect that DB2/2 will follow closely. ■

• IBM Corp., 800-342-6672.

PROTECTING DATA *(cont. from page 60)*
high-availability requirements and with a very competent administrator.

Oracle7. Oracle7 uses a multifile, static transaction log very similar to that of Informix Online. Oracle does not, however, store undo in its log like Online does, so the Oracle log does not limit transaction sizes.

Like Ingres, Oracle makes fuzzy online backups — Oracle does not degrade application performance during an online database backup, even in demanding systems. However, unlike other servers, Oracle does not have any process or mechanism that automatically copies files when you request a backup — an Oracle administrator must manually back up files using the operating system or a backup utility.

Oracle doesn't offer an incremental backup option, but does permit you to back up individual tablespaces (database partitions). This is useful if you want to back up active tablespaces more frequently than passive tablespaces. Oracle also allows you to back up multiple tablespaces in parallel, a nice feature for reducing backup time in large databases. To protect against single points of failure, Oracle allows you to mirror log files, but not data files.

Coupled with its tablespace backup capability, Oracle is the only server that allows you to restore and recover only the damaged files of a database. Oracle even lets you recover damaged tablespaces offline and in parallel while the remainder of a database is online. Oracle supports both complete and point-in-time, roll-forward recovery options.

Probably because of Oracle7's origin in high-end systems, its backup and recovery features are comprehensive and best-suited for the most demanding database systems. If you are using a small or lightly used system, you may find Oracle's offerings overkill and complex.

Sybase SQL Server 4.9. Sybase SQL Server 4.9 has a single-file transaction log, very similar to that of Ingres. SQL Server's single log file, however, grows dynamically as the server logs new transactions and shrinks dynamically when you archive and "truncate" old entries from the log. The ultimate size of the transaction log file depends on the disk space available where you place it. SQL Server does not have any option to archive the log automatically — the administrator must monitor and truncate the log manually (or write a special program to do so).

Like most other servers, SQL Server 4.9 makes consistent online backups that can detract from the performance of ongoing transaction processing. SQL Server 4.9 does not support incremental backups.

For fail-safe protection and high availability, SQL Server can mirror all files of a database. If you should need to recover a database, SQL Server requires that you restore the entire database from the most recent complete database backup. SQL Server supports complete offline roll-forward recovery, but does not have a point-in-time recovery option.

In summary, Sybase SQL Server 4.9 fits best with middle-size systems that require high-availability features. A SQL Server administrator must keep watch or design a program to ensure that the transaction log doesn't grow close to its limit. ■

• Gupta Technologies, 1060 Marsh Rd., Menlo Park, CA 94025; 800-876-3267, 415-321-9500, or fax 415-321-5471.

• Informix Software Inc., 4100 Bohannon Dr., Menlo Park, CA 94025; 800-331-1763, 415-926-6300, or fax 415-926-6593.

• Ingres Corp., 1080 Marina Village Pkwy., P. O. Box 4026, Alameda, CA 94501; 510-769-1400 or fax 510-748-2670.

• Oracle Corp., 500 Oracle Pkwy., Redwood Shores, CA 94065; 800-672-2531, 415-506-7000, or fax 415-506-7200.

• Sybase Inc., 6475 Christie Ave., Emeryville, CA 94608; 800-879-2273, 510-596-3500, or fax 510-658-9441.

Supplied by the British Library - "The world's knowledge" www.bl.uk

Home Search List First Prev Go to Next Last ☑ Include

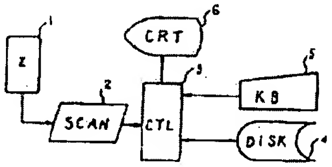
MicroPatent® PatSearch Fulltext: Record 1 of 1

Search scope: JP (bibliographic data only)
Years: 1981-1990
Patent/Publication No.: ((JP63010286))

Order This Patent | Family Lookup | Find Similar | Legal Status

Go to first matching text

JP63010286 A
IMAGE RETRIEVING SYSTEM
HITACHI LTD



Click here for larger image.

Abstract:

PURPOSE: To retrieve similar images at a high speed by having comparison between binarization input image data and its mirror reflected image data and registered image data respectively in the minimum necessary ones of those divided areas. CONSTITUTION: An image input device 2 reads the image information on a document 1 and sends it to a control part 3 after converting it into the black/ white binary data. The part 3 stores this binary data in its built-in picture memory as the 2-dimensional registered data and calculates the number of black dots for each of divided small areas to produce an index serving as a key for retrieval of the image data. These image data and index are registered to an external memory device 4. When a retrieving indication is received from a keyboard 5 and displayed on a CRT 6 for image data, the part 3 performs comparison between the binarization input image data and its mirror reflected and registered data respectively in the minimum and necessary ones of those divided areas and decides he coincidence of these data with a prescribed allowable range.

COPYRIGHT: (C)1988, JPO&Japio

Inventor(s):
KISHINO TORU

Application No. 61154003 JP61154003 JP, **Filed** 19860702, **A1 Published**

19880116

Int'l Class: G06F01570

Patents Citing This One No US, EP, or WO patent/search reports have cited this patent.

Home Search List First Prev Go to Next Last

For further information, please contact:
Technical Support | Billing | Sales | General Information

AT 2

INTERNATIONAL SEARCH REPORT

Intı ɔnal Application No
PCT/US 00/32493

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F11/14

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, INSPEC, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 901 327 A (OFEK YUVAL) 4 May 1999 (1999-05-04)	24
Y	abstract	1-5,7,8, 19-21, 25-32, 39-43
A	column 4, line 49 -column 7, line 65 column 8, line 51 -column 9, line 28 column 10, line 49 -column 11, line 16 column 35, line 7 - line 24 column 38, line 23 -column 39, line 4 column 47, line 13 -column 48, line 67 figure 1 --- -/--	9-11,33, 34

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search 1 August 2001	Date of mailing of the international search report 27/08/2001
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Leuridan, K

INTERNATIONAL SEARCH REPORT

Inte onal Application No
PCT/US 00/32493

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 92 18931 A (EASTMAN KODAK CO) 29 October 1992 (1992-10-29)	1-5,7,8, 19-21, 25-32, 39-43
A	abstract page 1, line 4 - line 9 page 8 -page 10, line 2 page 19, line 11 -page 23, line 18 figures 1,7,8 ---	24
A	VEKIARIDES N: "Fault-tolerant disk storage and file systems using reflective memory" PROCEEDINGS OF THE TWENTY-EIGHTH HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, PROCEEDINGS OF THE TWENTY-EIGHTH ANNUAL HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, WAILEA, HI, USA, 3-6 JAN. 1995, pages 103-113 vol.1, XP002173081 1995, Los Alamitos, CA, USA, IEEE Comput. Soc. Press, USA ISBN: 0-8186-6930-6 page 104, right-hand column, line 5 -page 105, left-hand column, line 22 page 107, left-hand column, line 17 -right-hand column, line 12 page 108, left-hand column, line 25 -right-hand column, line 2 page 109, left-hand column, line 20 -right-hand column, last line page 110, left-hand column, line -4 -right-hand column, line -3 figures 2,4,7 ---	1-4,7,8, 19,21, 24-32, 39-43
A	"PROGRAM FOR EXPORTING TRANSMISSION CONTROL PROTOCOL-BASED SERVICES THROUGH FIREWALLS" IBM TECHNICAL DISCLOSURE BULLETIN,IBM CORP. NEW YORK,US, vol. 40, no. 12, 1 December 1997 (1997-12-01), pages 161-162, XP000754125 ISSN: 0018-8689 the whole document ---	9-11,22, 34
A	US 5 819 020 A (BEELER JR DONALD E) 6 October 1998 (1998-10-06) abstract column 10, line 18 -column 11, line 10 column 13, line 7 -column 14, line 65 ---	7-21, 33-38

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte· ·nal Application No
PCT/US 00/32493

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5901327 A	04-05-1999	US 5742792 A	21-04-1998
		US 6173377 B	09-01-2001
		US 6052797 A	18-04-2000
		US 6044444 A	28-03-2000
		US 5889935 A	30-03-1999
WO 9218931 A	29-10-1992	EP 0536375 A	14-04-1993
		JP 5508506 T	25-11-1993
US 5819020 A	06-10-1998	US 5974563 A	26-10-1999

INTERNATIONAL SEARCH REPORT

ational Application No
PCT/US 00/42337

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L12/24

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L H04Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 767 427 A (DIGITAL EQUIPMENT CORP) 9 April 1997 (1997-04-09)	1,8,13
Y	page 6, line 27 -page 8, line 15	2-4,6, 9-11, 14-16
	page 28, line 15 - line 57 ———	

-/--

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

25 September 2001

Date of mailing of the international search report

09/10/2001

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL – 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Peeters, D

INTERNATIONAL SEARCH REPORT

ational Application No
PCT/US 00/42337

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L12/24

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L H04Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 767 427 A (DIGITAL EQUIPMENT CORP) 9 April 1997 (1997-04-09)	1,8,13
Y	page 6, line 27 -page 8, line 15 page 28, line 15 - line 57 ---	2-4,6, 9-11, 14-16

-/--

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search
25 September 2001

Date of mailing of the international search report
09/10/2001

Name and mailing address of the ISA
European Patent Office, P.B. 5818 Patentlaan 2
NL – 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer
Peeters, D

INTERNATIONAL SEARCH REPORT

ational Application No
PCT/US 00/42337

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5 559 958 A (DIDNER JONATHAN R ET AL) 24 September 1996 (1996-09-24) column 1, line 49 -column 3, line 33; figure 1 column 5, line 4 - line 44 column 8, line 14 - line 50; figure 2 column 9, line 50 - line 67; figure 2 column 33, line 34 -column 34, line 13; figure 6A column 200, line 1 - line 11; figures 9C,9D,13 column 202, line 5 - line 28; figures 13,14 column 212, line 29 - line 54; figures 9A,10-15 ---	2-4,6, 9-11, 14-16
A	DICK BANNISTER: "Compaq Storage Resource Manager" STORAGE NEWS - EVALUATOR GROUP, 'Online! vol. 2, no. 4, April 2000 (2000-04), XP002178293 Retrieved from the Internet: <URL:http://www.evaluatorgroup.com/English/Collaterals/Newsletter/200004_Newsletter.pdf> 'retrieved on 2001-09-24! page 5, right-hand column page 6, left-hand column -right-hand column -----	1-17

INTERNATIONAL SEARCH REPORT

Information on patent family members

ational Application No
PCT/US 00/42337

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0767427 A	09-04-1997	US 5345587 A	06-09-1994
		EP 0767427 A2	09-04-1997
		AT 160034 T	15-11-1997
		AU 3980093 A	19-08-1993
		AU 3980193 A	19-08-1993
		AU 3980293 A	05-08-1993
		AU 3980393 A	19-08-1993
		AU 3980493 A	05-08-1993
		AU 639416 B2	29-07-1993
		AU 4305289 A	02-04-1990
		DE 68928433 D1	11-12-1997
		DE 68928433 T2	16-04-1998
		EP 0441798 A1	22-03-1990
		JP 6502505 T	17-03-1994
		WO 9003005 A1	22-03-1990
		US 5475838 A	12-12-1995
		US 5557796 A	17-09-1996
		US 5608907 A	04-03-1997
		US 5832224 A	03-11-1998
		CN 1043810 A	11-07-1990
		JP 2230847 A	13-09-1990
		CN 1044176 A	25-07-1990
		CN 1044175 A	25-07-1990
		JP 2277153 A	13-11-1990
		CN 1045656 A	26-09-1990
		CN 1044174 A	25-07-1990
		JP 2236767 A	19-09-1990
		CN 1044719 A	15-08-1990
		JP 3062155 A	18-03-1991
US 5559958 A	24-09-1996	US 5471617 A	28-11-1995
		US 5828583 A	27-10-1998
		AT 198115 T	15-12-2000
		CA 2104421 A1	22-02-1994
		DE 69329743 D1	18-01-2001
		DE 69329743 T2	12-04-2001
		EP 0585082 A2	02-03-1994
		JP 2533066 B2	11-09-1996
		JP 6175955 A	24-06-1994
		AT 164242 T	15-04-1998
		CA 2071804 A1	25-12-1992
		DE 69224775 D1	23-04-1998
		DE 69224775 T2	06-08-1998
		EP 0520769 A2	30-12-1992
		JP 5257914 A	08-10-1993
		US 5367670 A	22-11-1994

AR3

Proceedings of the 28th Annual Hawaii International Conference on System Sciences — 1995

Fault-Tolerant Disk Storage and File Systems Using Reflective Memory

Nicos Vekiarides
Department of Electrical and Computer Engineering[1]
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Most replicated storage and file systems either take a specialized hardware approach or a software-oriented approach to fault tolerance. This paper describes a fault-tolerant disk storage and file system that falls in between the hardware and software categories. The system uses Reflective Memory to interconnect an array of standard computers comprising a massively parallel system. This architecture provides the basis for high-availability replicated file and storage systems with the performance and low overhead expected from specialized hardware while offering the modularity and scalability of a distributed system. In this paper, we describe the implementation of the fault-tolerant file and storage system to run large scale I/O-intensive applications, such as emulation of a stable storage DASD subsystem. Preliminary performance measurements indicate that selectively broadcasting regions of Reflective Memory allows for virtually no overhead over conventional systems for supporting replicated, distributed storage and file services.

1. Introduction

Most existing fault tolerant systems may be categorized as either hardware solutions that rely on redundant logic, lock stepped hardware, and specialized power supplies, or software solutions that rely on standard network protocols or clustering techniques. Both approaches have their shortcomings: the hardware approaches tend to fall behind the technology curve and become obsolete very quickly; software approaches tend to incur high overhead and are often limited by the speed of the network interconnect. Often, neither approach is able to offer high performance for I/O-intensive applications such as on-line transaction processing (OLTP).

Studies indicate that hardware faults are a minor cause of down time in OLTP systems [6]. Much of the time spent off-line can be attributed to other reasons such as:

- Power or other environment-related failures
- Software failures, upgrades or repairs
- Data reorganization
- Failures due to operator error

Most solutions are not sufficient to address all of these issues and provide adequate performance for OLTP applications. While power failures can be addressed through specialized power supplies, they do not provide disaster recovery. Hardware or software upgrades and repairs require a modularity allowing some hardware or software subsystems to be taken off-line, upgraded and brought back on-line transparently. Data reorganization, which involves redistributing data among storage devices or migrating data to new devices, has been supported by proprietary mainframes for many years [4]. Yet this capability is lacking from many of today's open systems. Failures due to operator error are difficult to address through hardware and software fault-tolerance, since such errors are not always possible to detect. Yet the prevention of and recovery from operator error are essential to any mission-critical system.

1.1 Motivation

The goal of the Reflective Memory approach described in this paper is to provide a high performance alternative to current fault-tolerant systems that is able to address the common causes of down time in OLTP system. Architecturally, this approach falls somewhere between the hardware and software categories: rather than specialized hardware, it uses standard hardware and software components to implement individual subsystems of an massively parallel processor (MPP); for performance and throughput, it uses a high-bandwidth Reflective Memory bus to interconnect the subsystems; for ease of integration, it uses standard memory interface as the basic communication mechanism.

Reflective Memory allows selected memory regions to be reflected or shadowed between two or more subsys-

[1] This research has been supported by Encore Computer Corporation, 6901 W. Sunrise Blvd, Ft. Lauderdale, FL 33313.

1060-3425/95 $4.00 © 1995 IEEE

tems comprising the MPP. This hardware-assisted memory reflection provides a fault-tolerant and persistent global shared memory mechanism; a subsystem can be individually removed or added while maintaining a globally consistent view of the shared-memory region and without disrupting memory accesses and updates already in progress from other subsystems.

The capability to dynamically remove, repair, and return subsystems of an MPP to service transparently, combined with the advantages of standard software that uses shared-memory as a basic communication mechanism has helped to implement a fault-tolerant file and storage system with very promising features:

- virtually no performance overhead for maintaining replicated storage and file systems
- on-line maintenance and software upgrades, where individual MPP subsystems can be removed or added without disrupting the remainder of the system
- disaster recovery capability allowing a single configuration to span several buildings
- on-line data reorganization and migration to maintain high utilization and efficiency
- efficient switchover between failed nodes (less than 1 second for detection and failover)
- on-line automated recovery and reconfiguration to reduce the possibility of operator error

These features have been implemented on the Encore Infinity 90 and have been used to run a non-stop IBM DASD [7] emulation application [13]. Preliminary experience and performance measurements indicate that Reflective Memory is a very efficient interconnect for building massively parallel fault-tolerant systems.

In the remainder of the paper, we compare Infinity to other fault-tolerant systems, highlight key design decisions and describe the user interface, enumerating the different configuration alternatives. We continue by describing the design, presenting an analysis of the algorithms used to implement mirroring services. Finally, we summarize our experiences with DASD emulation that has been shown to survive node failures and utilize on-line recovery in a manner transparent to the end user.

1.2 Overview

This paper describes a fault-tolerant disk storage and file system on the Infinity, a Reflective Memory-based MPP. Infinity is built from "off-the-shelf" nodes, each an independent computer with its own memory and power-supply, interconnected via the Reflective Memory System. Following an installation, certain memory areas are maintained consistent across all nodes. In these areas, an update by one subsystem is immediately visible across the entire MPP.

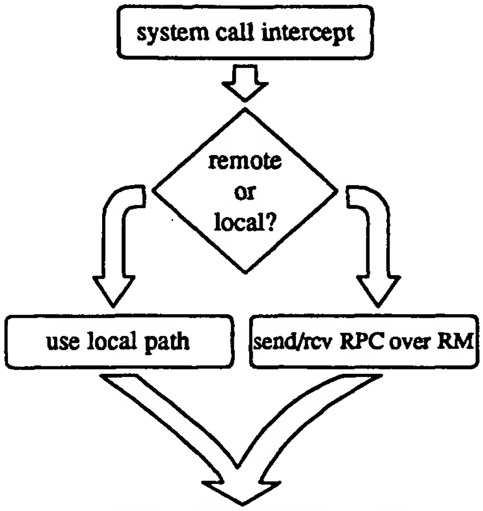


Figure 1: System Call Intercept

Fault-tolerance is based on the full or selective replication of a file system on at least two distinct subsystems called Input/Output Caching Controllers that, in conjunction with transaction processing systems, provide both a hardware and software fault-tolerant environment (transaction processing using system pairs results in software fault-tolerance [5]). Input/Output controllers run a standard UNIX operating system and provide cached file system or disk services on top of a massive array of disk controllers, drives and tape backup devices. User programs run on a distinct set of computing subsystems or a mainframe. These computing subsystems or nodes may also run a standard UNIX operating system and, although they may support a local file system, they maintain little state and offload most of the file-system work to the Input/Output controller. The computing node essentially performs only a routing function: as illustrated in Figure 1, it intercepts a system call, determines its destination, forwards the call to the appropriate controller which executes the file system task and returns the result.

Differently from network operating systems that may incur overhead in handling client/server protocols and associated data copying, Infinity relies on the "zero-protocol", shared memory-based communication that adds virtually no overhead to file system calls performed on the Input/Output controller. More importantly, it avoids data copying, a typical bottleneck in many UNIX file system implementations, by utilizing memory mapping: a dynamic configuration capability allowing blocks of memory to be either in private mode (mapped into a single program's address space) or in reflective mode (mapped into the network shared memory space). Combining the dynamic mapping capability with extremely efficient Remote Procedure Call (RPC) mechanism, the controllers act as an intelligent DMA engine capable of executing disk transfers directly into the user address space, without intervention from the operating system or user program. Figure 2 illustrates this memory-mapped I/O.

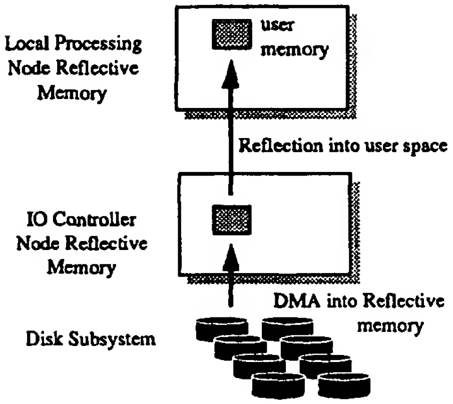

Local Processing Node Reflective Memory
user memory
Reflection into user space
IO Controller Node Reflective Memory
DMA into Reflective memory
Disk Subsystem

Figure 2: Memory-mapped I/O

The dynamic memory management capability has been used to implement a mirrored, almost no overhead fault-tolerant file and storage system that is the subject of this paper. Dependent upon configuration, a file system call may be forwarded to either one I/O controller or pair of replicated controllers. It is transparent to the end user program whether a write request is issued to one or two I/O controllers.

The controller replication is enabled by a mount command option that can be individually selected by the system administrator: some critical file systems may be replicated; others, to save memory and/or disk space, are not replicated. The flexibility to configure mirroring at mount time allows the same set of controllers to be used for both load balancing and replication. It is the decision of the system administrator whether or not to specify mirroring when installing a new file system.

In this paper we describe the implementation and experiences using the fault-tolerant file system to run large scale I/O-intensive applications. Since the fault tolerance is application transparent, no modifications were made to the applications. The overhead for maintaining fully replicated disk services has been measured to be less than 5%. The overhead of maintaining replicated caching for dual-hosted disk subsystems has been measured to be negligible due to the broadcast capability of Reflective Memory. The delay in switchover, in case of a controller failure, is unnoticeable (less than 1 second). On-line repair times depend on the type of failure. In the case of a disk media failure, replacing a disk and resynchronizing takes about 30 minutes depending on the disk capacity and bandwidth. In the case of a controller failure, recovery is proportional to the amount of time a controller has been out of service and the amount of updates occurring in that time period. The ceiling on this recovery time corresponds to the amount of time required to replicate all of the data on all of the disks using efficient parallel recovery algorithms.

1.3 Related Work

There is a wide spectrum of design alternatives in building fault-tolerant systems. At one extreme, hardware-oriented approaches are based on special-purpose, lock-stepped replicated hardware; at the other extreme, software approaches rely on routing and replication across a network to provide high-availability service to users. Infinity's architecture does not strictly follow either of the above categories.

Infinity is also different from the existing massively parallel computers, such as nCUBE [4] or Paragon [8], that support multiprocessing but fail to provide the shared-memory abstraction for their interconnects. In addition, many of those systems lack even rudimentary support for nodes being able to enter or exit the configuration without taking the entire system off line [17]. In contrast, Infinity's hardware and software allow any node to enter or exit the configuration without rebooting, atomically updating an entering node's state in Reflective Memory.

Infinity is not an "ultracomputer" in the quest for the Teraflops Supercomputer [3]. Instead, Infinity strives towards scaleable, fault-tolerant and massively parallel input/output processing [1] [12] [16]. From the user perspective, Infinity appears similar to the emerging "open-cluster" computers that are used for on-line transaction processing [13] [14].

Many of the "open cluster" computers use RAIDs [11] and dual-hosted or multi-hosted disk storage systems to increase reliability. IBM's AIX High Availability 6000 clusters utilize disk sharing between clustered workstations to provide higher availability [2]. Similarly, Infinity utilizes RAID technology and dual-hosted disks. Most such clustered systems, however, rely on sophisticated lock managers to coordinate file access between nodes. Unlike these systems, Infinity does not require a distributed lock manager because it runs only one file system that is accessible from multiple nodes. It also goes a step further than RAID in disaster recovery by offering remote disk replication over a fiber optic Reflective Memory link allowing replicated I/O controllers to be as far as 3 kilometers apart.

Infinity is conceptually similar to a shared-memory MPP except that the actual memory is shared via its "dynamic reflection" capability; memory is accessed and modified at hardware speeds, on the order of nanoseconds. Shared-memory based software mechanisms provide inter-process coordination and synchronization. In this paper we describe the software mechanisms which provide fault-tolerant input/output processing.

From the fault-tolerant input/output perspective, Infinity is different from the low-level approaches such as RAID in that it provides high-level mirroring. High-level mirroring allows for an entire file system, including cache, to be replicated, resulting in more flexibility and better performance. Moreover, it supports any file system since the file system resides at a lower level. For replicated caching with dual-hosted disk subsystems, Infinity uses mirrored caching in Reflective Memory. For full replication, Infinity's Log-Ahead algorithms bear similarity to other algorithms such as HARP [9], though the actual implementation is simplified by the capabilities of Reflective Memory.

1.4 Design Objectives

The main objective of the Infinity is to provide fault-tolerant file system and disk storage service to processing nodes interconnected via Reflective Memory. The fault-tolerant configuration offers the following features and functionality:

- support for mirrored disks and RAIDs with a unified file system
- automated on-line recovery
- no data loss from buffer cache on failures
- no overhead for mirrored buffer cache
- small overhead (5% or less) for replicated file system operations
- full utilization of system hardware, no standby modules
- fiber optic connection up to 3 km for disaster recovery
- supports all file systems and disks with no modifications to underlying file system or disk driver code

Key aspects of the dual controller system are the minimal performance overhead it imposes over conventional file systems and the ability to repair and recover the system on line, restoring it to full capacity without bringing the entire system down. On line repair and recovery is possible through the modularity of the Infinity controller, allowing subsystems and components to be added and removed without affecting the entire system.

In addition to fault-tolerance, utilization of redundant hardware during normal mode of operation effectively increases system capacity for transactions. In particular, transactions that update are broadcast to parallel controllers without very little or no degradation in performance; read transactions are routed among those controllers for better load balancing and higher overall I/O capacity.

In order to meet the above design objectives and simplify the implementation, the following constraints have been assumed:

- Recovery addresses single point of failure only. We assume the probability of the same component failure on multiple systems is low.
- Reflective Memory is available (minimum of 1 megabyte) for logging purposes

The goal of this paper is not to provide a panacea for all possible system failures of a disk storage or file system. The aim is to demonstrate an efficient model for building scaleable, highly available systems based on Reflective Memory.

2. Fault-Tolerant Architecture

The essential elements of the storage and file system are a pair of Input/Output cache controller nodes interconnected via Reflective Memory. We first present a brief overview of Reflective Memory and look at the various configuration options.

2.1 Reflective Memory

The Reflective Memory bus is the backbone of the Infinity computer, capable of interconnecting multiple processing nodes (or nodes) with at least two Input/Output controller nodes (or controllers) [14]. Each Reflective Memory bus provides a peak throughput of 53-100 megabytes per second. The high bandwidth of the bus, combined with a simple shared memory-based

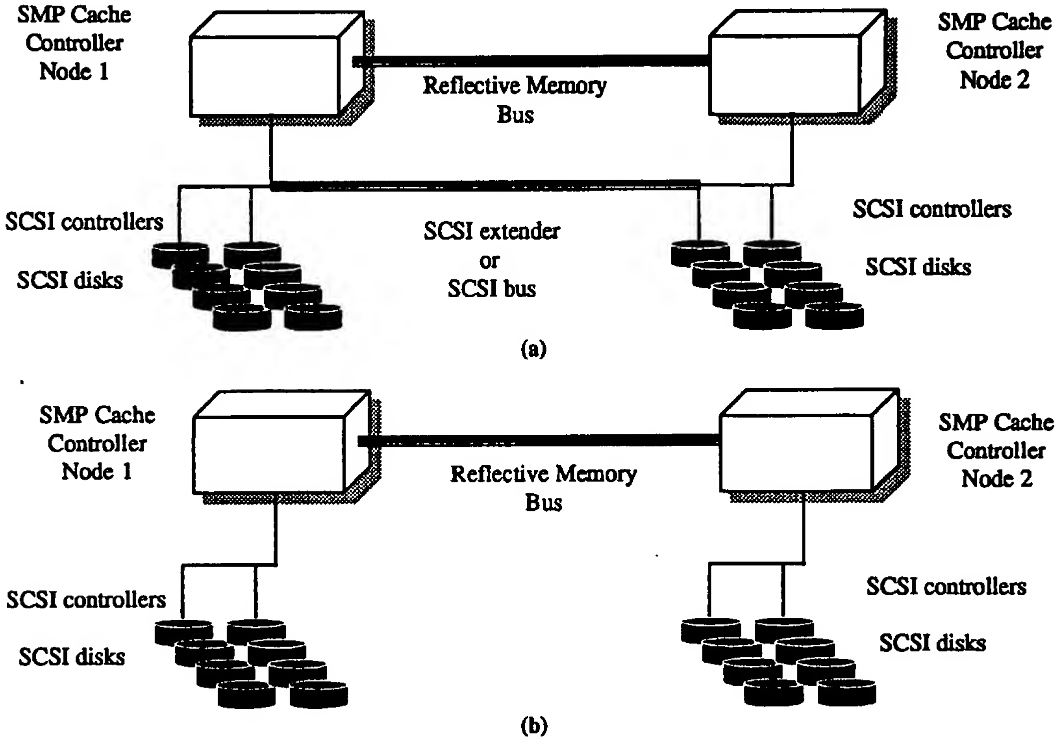


Figure 3: (a) Dual-hosted configuration (b) Fully Replicated Configuration

RPC, allows nodes to access the storage system residing on controllers at local speeds. A total of up to nine nodes and/or controllers can be configured to share Reflective Memory.

The actual Reflective Memory is a VME compatible board that contains 64 to 512 megabytes of memory to provide efficient coupling of processor nodes for time-critical applications. From an operating system point of view, it is an external memory board that is mapped in to an otherwise unused address range, with an access time of 75 nanoseconds. It behaves very similarly to a conventional memory board but allows memory updates to be selectively reflected across the Reflective Memory bus to other interconnected subsystems.

Although the Reflective Memory bus is a very high-bandwidth bus, it provides a reliable memory abstraction protected via parity checking. Network protocols with checksums are not necessary to ensure data integrity between communicating subsystems. Similarly, flow control is provided by hardware, maintaining the shared memory abstraction without affecting the way software is written. Utilizing shared memory principles offers "zero protocol" communication between physical memories on separate machines [10].

To provide replication spanning separate buildings, Reflective Memory supports a fiber-optic link (FORMS) allowing the bus to span distances up to 3 kilometers. The capability to physically separate the subsystems of the MPP provides a basis for supporting disaster recovery [8].

2.2 *Configuration Alternatives*

One of the main design objectives is to provide a flexible, user definable fault-tolerant configuration. This section describes two configurations, both providing transparent support for failure recovery and controller reconfiguration without directly interrupting service on processing nodes. Each has a different level of avail-

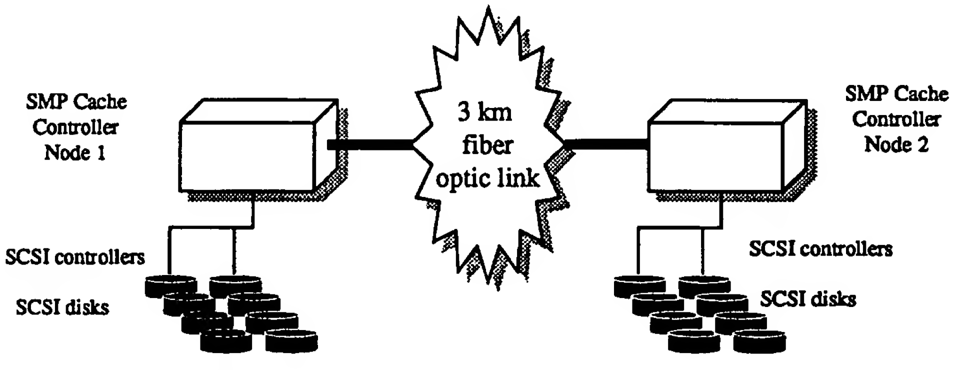


Figure 4: Fully replicated configuration using fiber-optic link

ability designed to meet the needs of various end applications:

- **dual-hosted configuration:** dual controller, single disk subsystem
- **fully replicated configuration:** dual controller, replicated disk subsystems

Figure 3 illustrates a dual-hosted configuration and the fully replicated configuration. Figure 4 shows a fully replicated configuration using FORMS.

Even the basic non-fault-tolerant configuration provides functionality to satisfy applications that cannot tolerate failures but can maintain pending I/O requests during the period of time a controller is down. This configuration emulates uninterruptible service by restoring all node state once a controller subsystem comes back up. File system caching maintained in non-volatile memory can be recovered following a failure. Since no error is returned to end application on the processing node, it may resume as if the failure never occurred.

Typical applications that take advantage of this configuration are automated applications, that do not impose the non-stop requirements of on-line transaction processing. If a controller failure is detected, all activity to that controller pauses, requests are kept pending and subsequently reissued once the controller is restored. This stop/start capability of the controller allows a controller subsystem to be taken down for repair or reconfiguration and restored, without having to restart the applications running on the processing nodes.

To enhance the reliability of the disk subsystem, mirrored disks or RAIDs may be used in conjunction with this configuration. Typically, on detection of a single disk failure, the controller subsystem can continue to function with the mirrored disk and recovered on line with a new disk.

The dual controller, dual hosted disk mirroring subsystem steps up reliability offering true uninterruptible service in the case of a single controller subsystem failure. Upon failure, all activity to that controller is rerouted to a backup controller and resumes seconds later in a manner transparent to the end applications. With a dual-hosted disk subsystem, both controllers share the same disks, requiring no file system resynchronization once a failed controller has been restored.

Unlike most dual-hosted systems, Reflective Memory does not limit the maximum number of processing nodes in the system to the amount of hosts that can be configured to use the disk subsystem. Multiple processing nodes can share the same dual-hosted controller system over Reflective Memory.

The fully replicated dual controller, separate disk subsystem offers the highest level of availability, providing uninterruptible file system service and data replication on physically separate media. As in the previous dual controller case, a controller subsystem failure causes requests to that controller to be rerouted to a backup. However, after restoring a failed controller, its file system is resynchronized by executing redo logs maintained on the other controller while it was down. Resynchronization does not require either controller to be taken offline, and, hence, occurs on-line, while updates are in progress from processing nodes.

In addition to on-line controller recovery, this system also offers on-line disk recovery. Similarly to a controller failure, requests for a failed disk are rerouted to the corresponding disk on the second controller transparently to the end application.

Table 1: Comparison of different levels of high availability controller

NODE	STORAGE	CONTROLLER HOT SWAP	DISK HOT SWAP	DISASTER RECOVERY
dual	single	yes	yes w/RAID	no
dual	replicated	yes	yes	yes

Meanwhile the system administrator may insert a new disk to replace the old one and initiate an on line recovery of the failed disk from the new disk.

"Hot swaps" of both disks and controller components are possible in this configuration with zero down time. For instance, a disk showing initial signs of failure such as multiple retries by the device driver can be replaced by a new disk and resynchronized on line.

The fully replicated configuration naturally provides a higher level of disaster recovery than do mirrored disk systems or disk arrays. Reflective memory may use a fiber optic link spanning up to 3 km. The dual controllers and disk subsystems could be located in separated rooms or even separate buildings using the fiber-optic link between the controllers.

2.3 DASD Configuration

In a DASD configuration, one or more Infinity controllers may be used to service mainframe requests. When two or more controllers are used, they comprise a fault-tolerant system that supports the following DASD functionality:

- dual-copy : disks may be mirrored between controllers over a fiber optic link
- concurrent copy : database snapshots can be backed up on line
- stable storage cached writes : I/O completes when it resides in Reflective Memory of two controllers

The DASD architecture, illustrated in Figure 5, allows a mainframe front end to communicate through its channel controllers and special MUX Channel boards. CKD simulation software allows emulation of 3380 type devices. This software is able to access the mirrored Infinity cache to issue I/O requests.

3. Implementation

To achieve mirroring, fault-tolerance and on-line recovery, the Infinity controllers make use of logging, sequencing and global data structures to provide on-line resynchronization. This section describes, in detail, the algorithms used for fault detection and recovery.

3.1 Health Monitor

A key aspect of fault detection is a non-intrusive health monitoring that determines the status of the entire MPP from each node. Changes in this status can trigger recovery events in the health monitor to provide uninterruptible service.

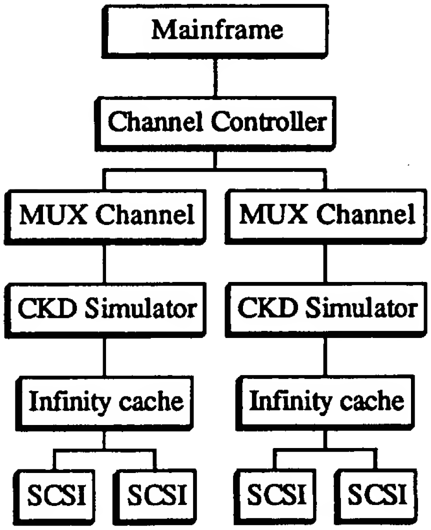


Figure 5: DASD Architecture

In brief, a global data structure is shared in Reflective Memory by all the nodes comprising the MPP. Each node is allocated a portion of this data structure in which it must increment a counter or "heart-beat". The status of other nodes can be monitored in their corresponding sections. The end result is the ability to detect node failures quickly so that requests can be rerouted to a back up node in a manner transparent to end application.

3.2 No-overhead Cache Replication

Using Reflective Memory, a replicated buffer cache can be maintained with no overhead over a conventional buffer cache. With the cache residing in Reflective Memory, all writes can be reflected to the cache memory on two controllers. On a dual-controller setup, the cache memory is able to survive any single point of failure on each controller.

Since each controller manages its own portion of cache memory, the mirrored controller need not do anything unless the other controller crashes, at which point it initiates recovery. Given that Reflective Memory imposes no overhead for broadcasting, there is no added latency associated with a replicated cache. No processing or bus cycles are lost in replication.

On a fully replicated disk setup, data needs to be flushed from the cache to a separate set of disks residing on separate controllers. In order to maintain consistent copies of both disks, a Primary Copy Log-Ahead mechanism is used on each controller. Before a buffer is flushed locally, the buffer descriptor and associated information are logged in memory and issued to the other controller. The buffer is then flushed locally. Once the remote controller returns completion, the buffer can then be replaced on the cache LRU for future use. If a controller fails, logs are kept on the active controller and replayed when the failed controller is restored. We describe this log-ahead algorithm in more detail in the following section.

Because of the asynchronous nature of writes to the cache, this logging does not introduce any added latency other than the logging of data in memory. To the end application, writes still complete asynchronously by virtue of residing in two controller memories.

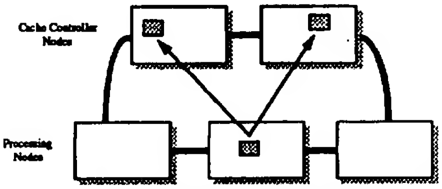


Figure 6: Broadcast reflection

3.3 File System Mirroring

The file system uses a Primary Copy Log-Ahead replication scheme where disks are replicated or shared on two controller subsystems. The disks are distributed in such a way so that one controller serves as a primary for half of the disks while the other controller is a secondary. All accesses that do not modify data occur only on the primary controller; updates of data occur on both the primary and secondary to maintain file system consistency. As shown in Figure 6 and mentioned previously, updates are broadcast to replicated controller nodes, introducing no added transmission latency.

The primary controller for a particular file system uses the Log-Ahead protocol to continuously log control data for each file system update using a circular buffer in Reflective Memory. Each log entry contains control data corresponding to a single disk update. The log-ahead protocol forces every update to enter the log before it is processed on either controller; this ensures that a record is kept of each change until both controllers acknowledge its completion.

A typical request to modify data is first logged on the primary controller's Reflective Memory, making it visible to the secondary controller; the request is then processed on the primary. Once the request completes on the primary, a completion is posted to the processing node (the end application), regardless of whether the secondary has also completed the request. When the request completes on the secondary, the secondary marks a status in the log and finally removes the log entry. Figure 7 demonstrates this log-ahead scheme.

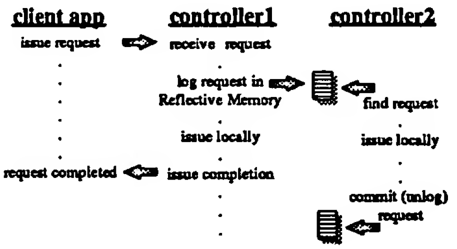


Figure 7: Log Ahead protocol

The log kept in Reflective Memory has four points used to ensure consistency. There is a log pointer (LP) indicating the next available log entry. A local commit point (LCP) indicates the last log entry which has been applied locally on this controller. A global commit point (GCP) indicates the last log entry that has been applied both locally and on the mirrored storage node. Finally the log base is the point below which log entries can be removed. Figure 8 indicates the points of the log.

Because of the asynchronous nature of updates between the replicated storage nodes, at any particular instant, the two controllers may be not be consistent between updates. To provide switchover upon failure that guarantees consistency between storage nodes, all requests for updates by the processing nodes are sequenced. Sequencing ensures that logs have been drained on the primary controller before the secondary replaces it. This scheme guarantees that the last update on the primary will always be accessible on the secondary.

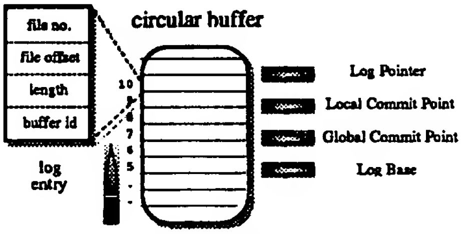


Figure 8: Recovery Log

Processing of all updates is guaranteed by the log-ahead protocol. If a secondary controller should go down and not receive an update, the log entry for that update will remain until it is reissued once the secondary is restored. Figure 9 illustrates the sequence of events following a controller failure.

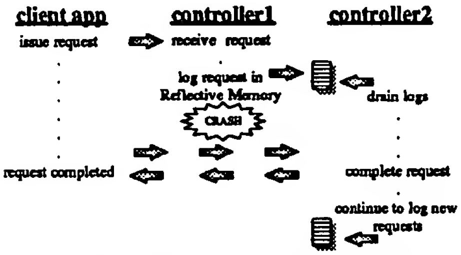


Figure 9: Recovery with log ahead

3.4 Recovery from Controller Failures

While a controller is down, its mirrored controller continues to log modifications in its circular log buffer. The size of the circular buffer is configurable; typically it may be configured to store up to 30 minutes of I/O updates for a moderately loaded system. Should the log buffer overflow, all affected disk subsystems must be copied and restored. As a rule of thumb, the log should be large enough to accommodate updates not exceeding the time it takes to copy each of the disks in parallel.

Assuming that the failed controller can be restored within a reasonable amount of time (less than the length of the log buffer), recovery occurs automatically with the active controller executing its recovery log; the log is drained while activity is still in progress to the active controller. If the time for recovery exceeds the length of the log buffer, a full disk subsystem on-line recovery is required. Ensuring a quick recovery period may involve slowing down incoming updates to a point that will not exceed the worst case recovery speed. This slowdown only occurs when recovery is in progress and adjusts dynamically depending on system load and log utilization.

Updates to a mirrored controller are asynchronous and may be issued very quickly. However, the logged updates only contain control data, and recovery involves local reads of data followed by remote writes. Since the remote cached writes complete asynchronously, local reads are sped up to keep pace. Batching reads, using a look-ahead technique on the log, achieves a speedup in recovery, minimizing resynchronization time.

Once recovery of a mirrored controller is complete, mirrored updates resume in the same fashion as before the failure. In addition, the recovered controller broadcasts a message to all of the other nodes in the cluster requesting that it be reinstated.

3.5 Recovery from Disk Failures and On-line Reconfiguration

When a disk failure is detected on one of the controllers under a fully replicated disk configuration, all I/O to that disk is rerouted to a backup controller, and a system message indicates the failure. The failed disk can then be "hot swapped" and recovered on-line. Once the disk is replaced, on line recovery may be initiated.

Unlike a recovery of an entire controller, where only logged modifications are required to be redone, the case of a disk failure generally requires cloning a new disk in its entirety. Since modifications to that disk could be in progress during the recovery period, the mirroring mechanism provides an "atomic update mode" for that mirrored disk pair (see Figure 10). This mode effectively ensures atomicity of updates to a particular disk across two controller subsystems while recovery is in progress. Using global device locking, a recovery process can resynchronize a new disk while updates are in progress.

When the new disk is fully cloned, the mirroring mechanism takes the disk out of "atomic update mode" and continues mirroring updates.

Highlights of the on-line recovery/reconfiguration include:

- Zero down time for single disk failures
- Can upgrade existing storage system on-line
- Can upgrade an Encore Infinity to a replicated Infinity on-line

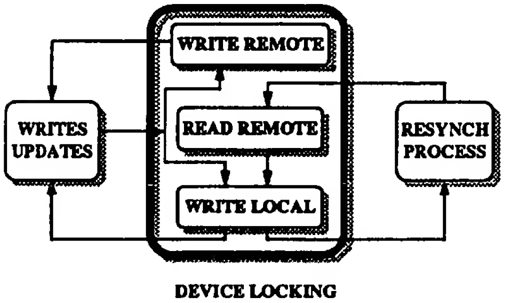


Figure 10: Coordination to recover or migrate data on-line

4. Experience and Summary

The goal of the Infinity design has been to support an open, fault-tolerant architecture for massively parallel input/output services. The Infinity architecture accommodates standard hardware and software components. Although the interconnect is the proprietary Reflective Memory bus, the shared memory interface runs applications that range from standard OLTP applications to DASD emulation software.

Preliminary experience with the fault-tolerant DASD configuration has indicated that the Infinity is able to provide fault tolerance to high performance I/O-based applications. In a dual controller setup, controllers can be added or removed from the configuration without affecting the behavior of end applications. Each node has a separate power supply, Reflective Memory and cabling. These components can be repaired while the rest of the system remains on-line. Disks or RAIDs can be shared by the controllers or replicated. Replication may span a fiber-optic link so that critical information may be replicated in separate locations to provide disaster recovery.

This system comprises the first step in achieving a fault-tolerant hardware and software environment. It is a fault-tolerant system that survives single component failures. Beyond fault-tolerance, the modularity of the system supports on-line repairs, software upgrades and data reorganization.

The DASD configuration uses a replicated cache in Reflective Memory with the option to have critical disk storage fully replicated. To conserve memory, a portion of the cache is used for write updates that is visible to both controllers while the remainder is used as read cache. Since the read cache portion is not required for recovery, it is not replicated. With a fully replicated disk residing on dual controllers, it is important that each controller make modifications that go to stable storage. For this case, the log-ahead algorithm is used.

Performance testing on a prototype replicated cache only system showed that cache updates performed at the speed of the Reflective Memory bus, and, as expected, showed no added latency since only one controller manages its portion of the cache.

Table 2: Overhead of Replicated Updates

Update Type	End User Completion	Added Latency
Normal	complete I/O	-
Replicated Cache Only	complete I/O	none
Replicated Cache and Disks	log & complete I/O	less than 5%

For replicated cache and disk systems or replicated UNIX file systems, the log-ahead protocol adds a small amount of latency to each disk or file system update. There is no added synchronization between the controllers, but each replicated update involves logging recovery data in Reflective Memory. The recovery information varies depending on the type of update that is being processed. Typical disk updates may require as little as 16 bytes of log data. More complex file system calls may require slightly more.

Based on preliminary tests issuing replicated I/Os, this log-ahead protocol has shown a maximum of 5% added latency (Table 2). This is a conservative figure based on null I/Os. Actual I/Os make the added latency relatively smaller.

Infinity has also achieved the on-line recovery and reconfiguration goals by allowing an individual node to be taken off-line without affecting the rest of the configuration. This capability to reboot an individual node has been used not only to recover from CPU or memory failures but also to recover from software errors which often are followed by software upgrades. All of this has been done at run-time, without affecting the rest of the nodes. Furthermore, failures of the Infinity controller have been virtually unnoticed by Infinity users who have only experienced a slightly degraded response time due to the heavier load on the remaining controllers.

Failures of individual disks are also handled on-line. Having detected a failure of a disk, the operator replaces it with a new one and starts a disk-to-disk copy utility. This utility runs the disk in "atomic update" mode ensuring consistency of the primary disk and the secondary disk. Because recovery is synchronized with incoming updates, the elapsed time for getting the secondary disk back on line is not significantly delayed.

Overall, our experiences have indicated that flexibility is a main feature of the Infinity architecture, allowing configuration choices that best meet the needs of end applications. The goal has been to provide a massively parallel computer built out of standard components with clearly defined redundancy choices.

As a side effect of our development, we have learned that Reflective Memory when utilized as a shared-memory abstraction is a very powerful programming tool, able to significantly speed up the development cycle. Multiple Infinity nodes and controllers have been emulated on symmetric multiprocessors as UNIX processes having access to shared memory. Because Reflective Memory follows standard shared memory semantics, these emulations were subsequently moved into actual systems with little programming effort.

5. Conclusions

Reflective memory interconnected computers are well suited to building large scale fault-tolerant disk storage and file systems for I/O intensive applications. The modularity of both computational nodes and input/output controller nodes allows nodes to be removed or added to a Reflective Memory configuration without any down time in the remaining nodes. This modularity in a massively parallel architecture yields a very robust disk storage and file system. Preliminary experience with the system indicates that the broadcast capabilities of the Reflective Memory allow redundant operations, critical to achieving fault tolerance, to occur at speeds approaching conventional systems.

REFERENCES

[1] *AIM Performance Report*, AIM Technology, Santa Clara, CA 95054.

[2] *AIX High Availability Cluster Multi-Processing/6000*. International Business Machines Corporation, 1992.

[3] Bell. Ultracomputers, a Teraflop before its time, *Comm. of the ACM*, August, 1992.

[4] Erik DeBenedictis. *nCUBE Parallel I/O Software*, nCUBE Corporation, 1992.

[5] Gray and A. Reuter. *Transaction Processing Concepts and Techniques*, Morgan Kaufman Publishers, Inc., 1993.

[6] J. Gray and D. Sieworek. High-Availability Computer Systems, *IEEE Computer*, September, 1991.

[7] J. Hennesy and D. Patterson. *Computer Architecture A Quantative Approach*. Morgan Kaufman Publishers, Inc., 1990.

[8] *Intel Paragon XP/S Supercomputer Spec Sheet*. Intel Corporation.

[9] Liskov et. al. *Harp File System*, MIT Technical Report, 1992.

[10] *Memory Channel II*. Encore Computer Corporation, Publication no. 307-2468, 1994.

[11] D.Patterson, G.Gibson. *A Case for redundant arrays of inexpensive disks (RAID)*, University of California, Berkeley, 1988.

[12] Pieper. *Parallel I/O Systems for Multicomputers*, CMU-CS-89-143, June, 1989.

[13] Reese and K. Stukenborg. Implementing Highly Available Oracle Solutions, *Proc. International Oracle User Week*, October, 1993.

[14] Reflective Memory Patents, United States Patent, No.4,991,079, February 5, 1991. Continuation of No. 710,229, March 11, 1985.

[15] Slingwine, M. Sweiger. Node Recovery in the Sequent Distributed Lock Manager, *Journal of Open Systems*, Spring, 1993.

[16] Transaction Processing Performance Council, *TPC Quarterly*, San Jose, CA 1112-6311,

[17] Zajcew, O. Roy, D. Black and et. al. An OSF/1 UNIX for Massively Parallel Multicomputer, *1993 Winter USENIX*, January 25-29, 1993, San Diego, CA.

AS3

Implementation of a Fault-Tolerant Disk Storage System Using Reflective Memory[1]

Nicos Vekiarides
Department of Electrical and Computer Engineering
Carnegie Mellon University[2]

M.S. Project
Advisor: Dan Siewiorek

Abstract

Many replicated disk storage and file systems take either a hardware-oriented approach or a software-oriented approach to fault tolerance. This paper involves the implementation of a fault-tolerant disk storage system that falls in between the hardware and software categories. The implementation enhances a Reflective Memory-based massively parallel storage system consisting of an array of standard computers. Utilizing this architecture as a baseline, this project provides the basis for high-availability replicated file and storage systems without compromising the performance level achieved by the non-replicated systems. This paper entails the implementation of the log-ahead replication algorithms developed to run I/O-intensive applications such as a direct access storage device (DASD) subsystem. Performance measurements indicate that using a log-ahead primary copy replication algorithm in Reflective Memory imposes very little overhead (approximately 5% or less) over conventional systems for supporting replicated, distributed, cached storage system services.

1. Introduction

Many existing fault tolerant systems may be categorized as hardware solutions that rely on redundant logic, lock stepped hardware, and specialized power supplies, or software solutions that rely on standard network protocols or clustering techniques. Both approaches have their shortcomings. The hardware approaches are designed to scale to high workloads but tend to fall behind the technology curve and become obsolete quickly; they rarely use standard hardware components. Software approaches offer more flexibility, in choice

[1] Portions of this paper appear in the 28th Hawaii International Conference on System Sciences, January 3-6, 1995.

[2] This research was sponsored by Encore Computer Corporation, 300 Nickerson Road, Marlborough, MA 01752.

of components and configurations, but tend to incur significant performance overhead and are often limited by message passing, standard protocols and the speed of network interconnects. It is often difficult, using either approach, to offer high performance for I/O-intensive applications such as on-line transaction processing (OLTP).

Studies indicate that hardware faults are a minor cause of down time in OLTP systems [7]. Apart from power failures which may be addressed through specialized power supplies, system outages can be attributed to other causes such as:

- Software failures, upgrades or repairs
- Data reorganization, migration or backup

Hardware or software upgrades and repairs require a modularity allowing some hardware or software subsystems to be taken off-line, upgraded and brought back on-line transparently. Data reorganization, which involves redistributing data among storage devices or migrating data to new devices, has been supported by proprietary mainframes for many years, yet is not found in many open systems to date [6].

Data mirroring algorithms, such as primary copy replication, have been developed for transaction oriented systems to provide high availability and recovery from system failure [10]. While such algorithms are prevalent among fault-tolerant network file systems, often there is a significant performance penalty associated with replicated I/O at network speeds.

The Infinity architecture, outlined in this paper, provides the fundamental building blocks to realize a fault-tolerant system offering the advantages of the hardware and software approaches. A massively parallel processing (MPP) system, it interconnects nodes via high-bandwidth Reflective Memory, where each individual node consists of a computer built from standard components. This paper addresses the implementation of a fault tolerant disk storage system using the Infinity MPP.

1.1 Goals

The goal of this paper is to implement a high performance fault-tolerant file and storage system that is able to address many of the common causes of down time in OLTP systems through replication. The goal of the implementation is to keep the performance of the replicated system as close to the performance of the conventional system as possible during normal mode of operation. Further goals include extending the functionality of the system to include capability for on-line recovery and repair in order to minimize down time.

Infinity represents a modular MPP interconnecting multiple nodes. The Reflective Memory interconnect exploits the modularity of this architecture allowing to dynamically remove, repair, and return subsystems of an

MPP to service transparently without affecting the remainder of the system. These capabilities provide the basis for extending Infinity with fault-tolerant functionality.

Utilizing the basic Infinity MPP disk storage system, the objective is to implement a fault-tolerant file and storage system with the following features:

- minimal performance overhead for maintaining replicated storage and file systems
- capability for on-line maintenance and software upgrades, where individual storage subsystems (or MPP nodes) can be removed or added without down time
- capability for on-line data reorganization and migration to maintain high utilization and efficiency

A running prototype system will be used to evaluate these design goals. The implementation platform for the fault-tolerant system is a dual node Encore Infinity series. The primary end application is a non-stop DASD emulation application driven by a mainframe [8]. Preliminary experience and performance measurements indicate that Reflective Memory is a very efficient interconnect for building massively parallel fault-tolerant systems. We use the prototype DASD system to validate the low performance overhead of this replication scheme and to demonstrate the ability to recover the system and carry out data reorganization on-line.

In order to meet the design objectives and simplify the implementation, the decision is made to address single points of failure only. The probability of the same component failure on multiple systems is low and, therefore, the choice is made not to expend hardware resources beyond dual module redundancy to guard against concurrent failures.

Note that the goals of this paper do not including assessing the fault coverage of the implementation with respect to all possible system failures encountered by disk storage or file systems. The aim is to demonstrate an efficient implementation for building scaleable, highly available systems based on Reflective Memory and well-documented replication techniques. The low performance overhead of the implementation will be validated through performance measurements on a working system.

In the remainder of the paper, we compare Infinity to other fault-tolerant systems, highlight key design decisions and describe the architecture and the actual configuration used. We continue by describing the design, presenting an analysis of the algorithms used to implement mirroring services. Finally, we summarize our experiences with DASD emulation that has been shown to survive node failures and utilize on-line recovery in a manner transparent to the end user.

1.2 The Infinity Architecture

Architecturally, Infinity links standard hardware and software components to implement individual subsystems of a massively parallel processor (MPP). Each node of the system contains multiprocessor CPU board, local memory and a separate power supply. For performance and throughput, the high-bandwidth Reflective Memory bus interconnects the nodes or subsystems. For ease of programming and integration, it uses a shared memory abstraction as the basic communication mechanism and runs a standard UNIX operating system. Figure 1 illustrates the system architecture.

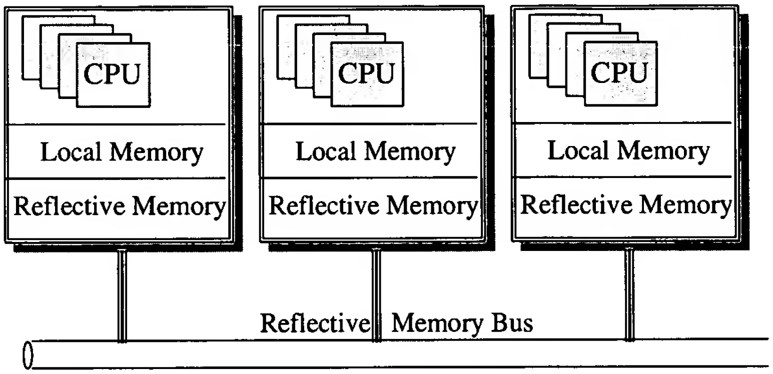


Figure 1: Infinity Architecture

Reflective Memory allows selected memory regions to be reflected or shadowed between two or more subsystems comprising the MPP. This hardware-assisted memory reflection provides a fault-tolerant and persistent global shared memory mechanism. The properties of Reflective Memory allow a subsystem to be individually removed or added while maintaining a globally consistent view of the shared-memory region and without disrupting memory accesses and updates already in progress from other subsystems.

The fault-tolerant storage system is based on the full or selective replication of a file system on at least two distinct subsystems, which we call Input/Output Caching Controllers, that, in conjunction with transaction processing systems, provide both a hardware and software fault-tolerant environment. Each Input/Output controller is capable of providing cached file system or disk services on top of a massive array of disk controllers, drives and tape backup devices. User programs may access the Input/Output controllers from a distinct set of computing subsystems or a mainframe. These computing subsystems or nodes may also run a standard UNIX operating system and, although, in some instances, may support a local file system, they maintain little state and offload most of the file-system work to the Input/Output controller. The front-end of the storage system essentially performs only a routing function. It determines the destination of each disk request and forwards the call to the appropriate controller which executes the task and returns the result.

Differently from network operating systems that may incur performance overhead in handling client/server protocols and associated data copying, Infinity relies on the "zero-protocol," shared memory-based communication that adds virtually no overhead to file system I/O performed on the Input/Output controller. More importantly, it avoids data copying by utilizing memory mapping: a dynamic configuration capability allowing blocks of memory to be either in private mode (mapped into a single program's address space) or in reflective mode (mapped into the network global shared memory space). Combining the dynamic mapping capability with extremely efficient memory based message passing, the controllers act as an intelligent direct memory access (DMA) engine capable of executing disk transfers directly into memory that can be accessed by multiple I/O controllers, without added intervention from the operating system aside from specifying reflection addresses at system initialization. Figure 2 illustrates memory-mapped disk I/O between controller nodes.

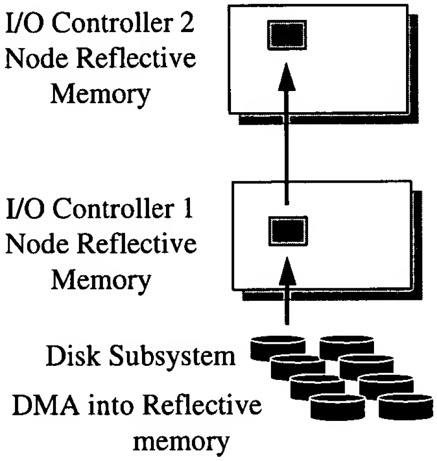


Figure 2: Memory-mapped I/O

Dynamic memory reflection implements a mirrored, low performance overhead, fault-tolerant cache for the disk storage system described here. Whether a file system call is forwarded to one I/O controller or pair of replicated controllers is dependent on the configuration of the Reflective Memory mapping. It is transparent to the end user program whether a write request is issued to one or two I/O controllers.

Because mirroring can be configured in hardware, utilization of redundant hardware during normal mode of operation can be used to effectively increase system capacity for transactions. In particular, a fault tolerant configuration broadcasts transactions that update to multiple controllers with very little or no degradation in performance; read only transactions are routed among those controllers. In either case, hardware reflection frees up replicated processing resources on each subsystem for local use until one of the subsystems experiences a failure.

1.3 Related Work

There is a wide spectrum of design alternatives in building fault-tolerant file and disk storage systems. At one extreme, hardware-oriented approaches are based on special-purpose, lock-stepped replicated hardware; at the other extreme, software approaches rely on routing and replication across a network to provide high-availability service to users. As a massively parallel processor, Infinity's architecture does not strictly follow either of the above categories.

Infinity is also different from the existing massively parallel computers, such as nCUBE [5] or Paragon [9], that support multiprocessing but fail to provide the shared-memory abstraction for their interconnects. In addition, many of those systems lack even rudimentary support for nodes being able to enter or exit the configuration without taking the entire system off line [19]. In contrast, Infinity's hardware and software allow any node to enter or exit the configuration without rebooting, atomically updating an entering node's state in Reflective Memory.

Infinity is not an "ultracomputer" in the quest for the Teraflops Supercomputer [4]. Instead, Infinity strives towards scaleable, fault-tolerant and massively parallel input/output processing [1] [13] [17]. From the user perspective, Infinity appears similar to the emerging "open-cluster" computers that are used for on-line transaction processing [13] [14].

Many of the "open cluster" computers use RAIDs [12] and dual-hosted or multi-hosted disk storage systems to increase reliability. IBM's AIX High Availability 6000 clusters utilize disk sharing between clustered workstations to provide enhanced availability [2]. Similarly, Infinity utilizes RAID technology and dual-hosted disks. Most such clustered systems, however, rely on sophisticated lock managers to coordinate file access between nodes. Unlike these systems, Infinity does not require a distributed lock manager because the file system resides on each I/O controller, not on the machine issuing requests.

Infinity is conceptually similar to a shared-memory MPP except that the actual memory is shared via its "dynamic reflection" capability; memory is accessed and modified at memory access speeds, on the order of tens of nanoseconds. Shared-memory based software mechanisms provide inter-process coordination and synchronization. This paper describes the software mechanisms that provide fault-tolerant input/output processing.

From the fault-tolerant input/output perspective, the replicated disk storage systems introduced here are different from the low-level approaches such as RAID in that they provide higher-level mirroring. High-level mirroring allows for an entire file system, including cache, to be replicated, resulting in more flexibility and better performance. Moreover, any file system can be layered on top of the mirrored caching in Reflective Memory. For full disk replication, the primary copy replication with log-ahead algorithm that will be detailed later bears similarity to other primary copy algorithms such as HARP [10], though the actual implementation is simplified by the capabilities of Reflective Memory.

2. Fault-tolerant Architecture

The essential elements of the fault tolerant storage and file system are a pair of Input/Output cache controller nodes interconnected via Reflective Memory. We first present a brief overview of Reflective Memory and look at the actual configuration.

2.1 Reflective Memory

The Reflective Memory bus is the backbone of the Infinity computer, capable of interconnecting multiple processing nodes with at least two Input/Output controller nodes (or controllers) [15]. Each Reflective Memory bus provides a peak throughput of 53-100 megabytes per second. The high bandwidth of the bus, combined with simple shared memory-based messaging, allows nodes to access the storage system residing on controllers at local speeds. A total of up to nine nodes and/or controllers can be configured to share Reflective Memory.

The actual Reflective Memory is a VME bus compatible board that contains 64 to 512 megabytes of memory to provide efficient coupling of processor nodes for time-critical applications. From an operating system point of view, it is an external memory board that is mapped in to an otherwise unused address range, with an access time of 75 nanoseconds. It behaves very similarly to a conventional memory board but allows memory updates to be selectively reflected across the Reflective Memory bus to other interconnected subsystems.

Although the Reflective Memory bus is a very high-bandwidth bus, it provides a reliable memory abstraction protected via parity checking. Network protocols with checksums are not necessary to ensure data integrity between communicating subsystems. Similarly, flow control is provided by hardware, maintaining the shared memory abstraction without affecting the way software is written. Utilizing shared memory principles offers "zero protocol" communication between physical memories on separate machines [11].

To provide replication spanning separate buildings, Reflective Memory supports a fiber-optic link (FORMS) allowing the bus to span distances up to 3 kilometers. Across the fiber-optic link, the semantics of Reflective Memory remain identical.

2.2 The DASD Configuration

In the DASD configuration, one or more Infinity controllers service mainframe requests. Figure 3 illustrates the non-replicated DASD configuration. In this configuration, a mainframe issues I/O requests to the Infinity controller through specialized multiplexing boards. The controller handles the request using a cache implemented in software and SCSI disks.

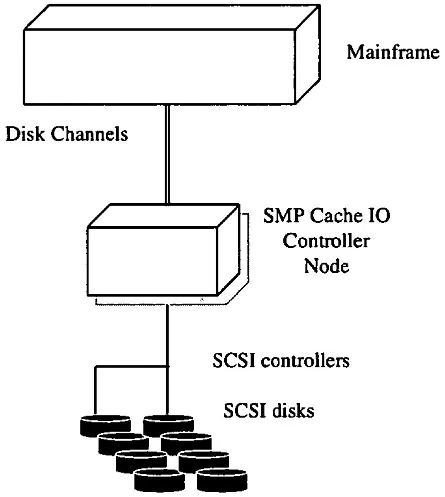


Figure 3: DASD non-replicated configuration

Figure 4 illustrates the replicated configuration where two controllers service mainframe requests. In this case, the I/O controllers are interconnected via Reflective Memory and the cache resides in Reflective Memory so that it is visible to both nodes. The fault-tolerant configuration can support the following high availability DASD functionality:

- dual-copy : disks may be mirrored on individual controllers or between controllers
- concurrent copy : database snapshots can be taken for backups on line

stable storage cached writes : I/O completes when it resides in Reflective Memory of two controllers and can subsequently survive the failure of a single controller

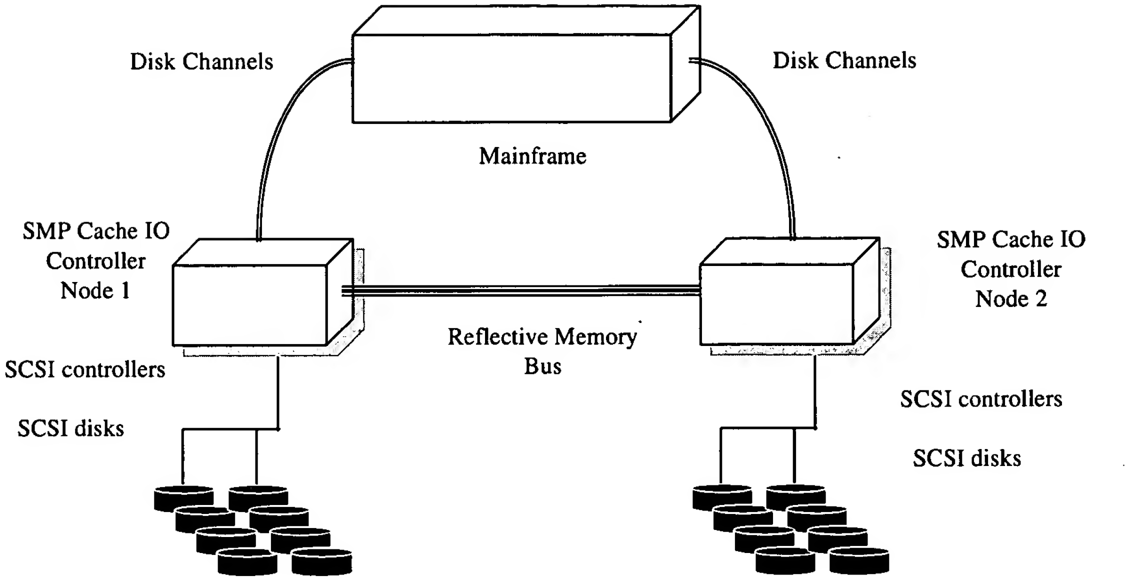


Figure 4: Replicated DASD configuration

The replicated DASD architecture and its associated hardware and software layers are illustrated in Figure 5. Note that besides the cache layer there is a count key data (CKD) emulation layer that implements emulation of 3380 type devices. This emulation software is able to access the mirrored cache and issues I/O requests.

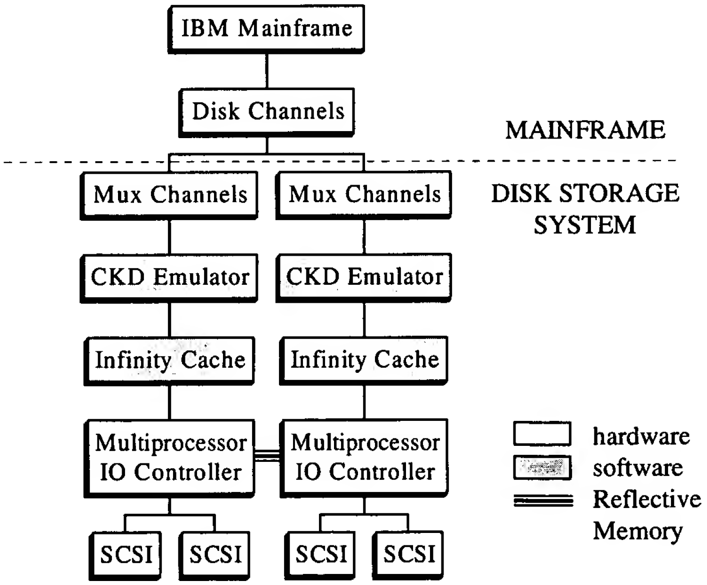


Figure 5: Replicated DASD Architecture

Note that upon failure of one the controllers, the logic to reissue a request to the second controller lies with the mainframe. Once the request has been issued to the second controller, the cache coherence logic that services the request correctly resides in the cache layer.

3. Implementation

Two software enhancements were implemented to support mirroring and on-line recovery for replicated systems: a primary copy log-ahead algorithm and disk synchronization across Reflective Memory. This section describes the implementation of these software enhancements.

3.1 No-overhead Cache Replication

In Reflective Memory, a replicated buffer cache can be maintained with no overhead over a conventional buffer cache. With the cache residing in Reflective Memory, all writes are reflected to the cache memory on two controllers, while reads remain local to each controller. Placing the write cache in broadcast memory allows all controllers access to the data.

Since each controller manages its own portion of cache memory and this implementation does not involve moving cache control structures into Reflective Memory, the write portion of the cache is subdivided between controllers. Each controller only accesses its own portion of the cache except in the case of a failure. With two controllers available, one serves as the primary for half of the disks while the second serves as a primary for the other half of the disks. For the mirrored half of the disks that each controller services, it need not do anything unless the other controller crashes, at which point it takes over. No processing cycles are wasted in cache replication.

Since the goal is to support a fully replicated disk setup, data eventually needs to be flushed from the cache to a separate set of disks residing on each controller. In order to maintain consistent copies of both disks, a Primary Copy Log-Ahead mechanism is used on each controller. Before a buffer commits locally, the buffer descriptor and associated information are logged in Reflective Memory in an area visible to the other controller. The buffer is then flushed locally. Once the remote controller returns completion, the buffer can then be replaced on the cache LRU for future use. If a controller fails, the secondary can determine what data still needs to be flushed by examining the logs that the failed controller has maintained in Reflective Memory. We describe this log-ahead algorithm in more detail in the subsequent section.

Because writes complete once their data is in memory, full replication does not introduce any added latency other than the logging of data in memory in maintaining a write-back cache. It does, however, provide a more stable volatile storage since data now resides in the memory of two controllers before going to stable storage. To the end application, writes complete by virtue of residing in two controller memories.

3.2 Full Replication with Log-Ahead

To provide replicated I/O at speeds of non-replicated systems, we use a Primary Copy Log-Ahead replication scheme between the two controller subsystems. The primary controller for a particular set of disks uses the log-ahead protocol to continuously log control data for each file system update using a circular buffer implemented in Reflective Memory. Each log entry contains control data corresponding to a single disk update. The log-ahead protocol forces every update to enter the log before it is processed on either controller; this ensures that a record is kept of each change until both controllers acknowledge its completion.

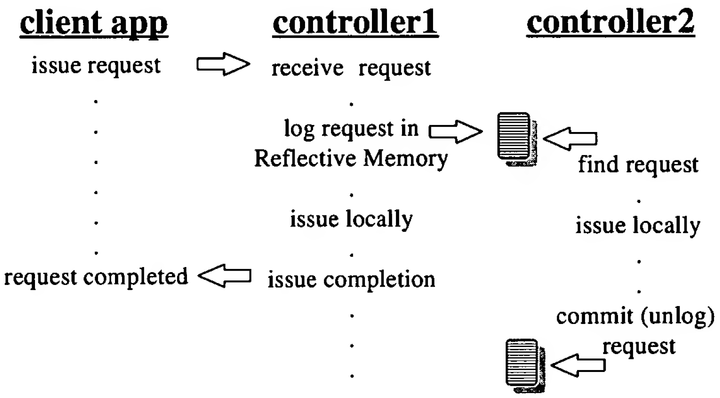


Figure 6: Log Ahead protocol

A typical request to modify data is first logged on the primary controller's Reflective Memory, making it visible to the secondary controller; the request is then processed on the primary. Once the request completes on the primary, a completion is posted to the end application, regardless of whether the secondary has also completed the request. When the request completes on the secondary, the secondary marks a status in the log and finally removes the log entry. Figure 6 demonstrates this log-ahead scheme.

To keep track of the status of updates between the two controller nodes, the log maintained in Reflective Memory has four consistency points. There is a log pointer (LP) indicating the next available log entry. A local commit point (LCP) indicates the last log entry which has been applied locally on a controller. A global commit point (GCP) indicates the last log entry that has been applied both locally and on the mirrored controller. Finally the log base is the point below which log entries can be removed. Figure 7 illustrates the points of the log.

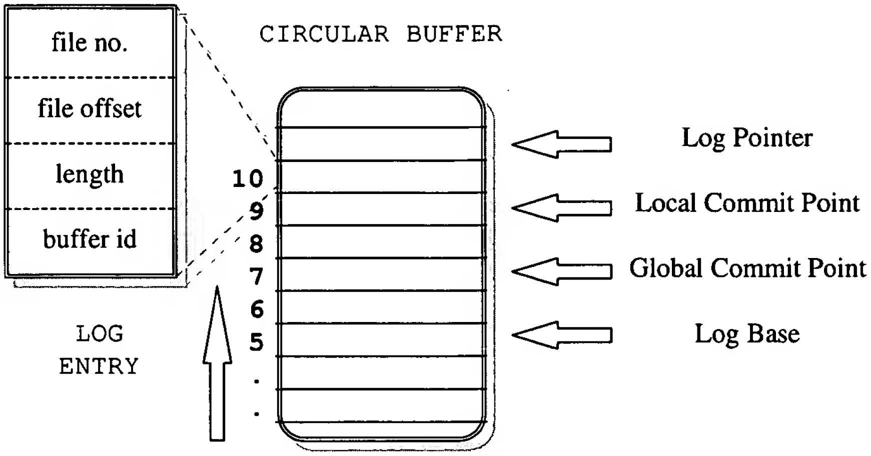


Figure 7: Recovery Log

Because of the asynchronous nature of updates between the replicated controllers, at any particular instant, the two controllers may not be consistent between updates. To provide failover guaranteeing consistency between controllers, all requests for updates by the processing nodes are sequenced. Sequencing ensures that logs have been drained from the primary controller before the secondary replaces it and begins to service new requests. This scheme guarantees that, for a single failure, the last update on the primary will always be accessible on the secondary.

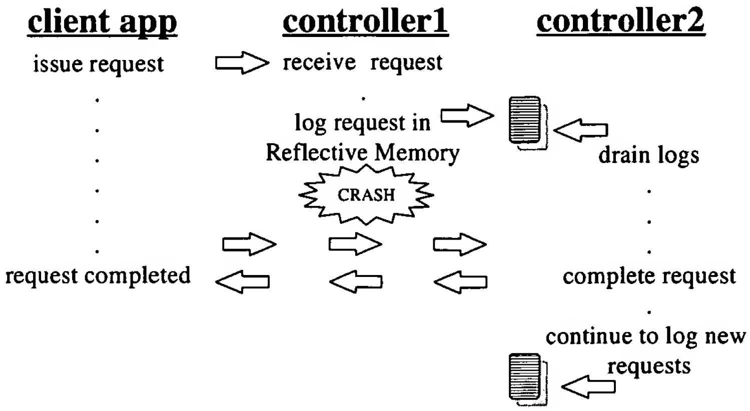


Figure 8: Recovery with log ahead

Processing of all updates is guaranteed by the log-ahead protocol even after one of the controllers fails. If either controller should go down and not receive an update, the log entry for that update on the other controller will be read so that the I/O is issued. Figure 8 illustrates the sequence of events following a controller failure. Note that once a primary controller goes down, the primary controller's logs need to be drained before the secondary can begin servicing requests.

3.3 Recovery from Disk Failures and On-line Reconfiguration

In implementing recovery from disk failures and the capability for on-line reconfiguration, the primary goal was to allow access to data at all times, leaving performance as a secondary concern. The assumption is made that suboptimal performance can be tolerated for a short period of time, given the length of time required to fully resynchronize a disk is relatively small (copying 1 gigabyte disks with a 1.5 Mbyte data transfer rate would require approximately 22 minutes).

When a disk failure is detected on one of the controllers under a fully replicated disk configuration, all I/O to that disk is rerouted to the backup controller, and a system message indicates the failure. The failed disk can then be recovered by replacing and by reinitiating the dual copy operation. During the recovery period, the system is able to accept updates and service read requests from the available disk containing the second copy of the data.

Allowing modifications to a disk during the period which recovery is in progress is achieved by inter-node coordination mechanisms. The implementation of this coordination uses locking across Reflective Memory to control access to the mirrored disk pair. This synchronization effectively ensures atomicity of updates to a particular disk across two nodes.

As Figure 9 indicates, a lock is used to serialize access to the device pair while recovery or synchronization is in progress. Assuming two nodes, one local and one remote, every incoming write must take a lock on the device pair while the data is written out to both devices. Concurrently, the recovery process must take a lock on the device pair for each read from the remote disk and hold the lock until it has finished the subsequent write to the local disk. This scheme allows full access to data while synchronization is in progress, albeit at the cost of serialized access and contention with the recovery process for the device lock.

Once disk resynchronization process completes, the locking requirements for each I/O are released and mirroring updates begin again using primary copy log-ahead. Note that this process applies to disks mirrored on a single node as well as between two nodes.

The significance of the on-line recovery mechanism is most apparent in that it allows the following features:

- Zero down time for a single disk failure and recovery

- Data can be migrated to new devices on-line by dual-copy enabling a set of new disks and disconnecting the old disks upon completion
- Data snapshots can be taken by dual-copy enabling a set of backup disks, or tape, and disconnecting them for a copy that is time-consistent at the instant of disconnect

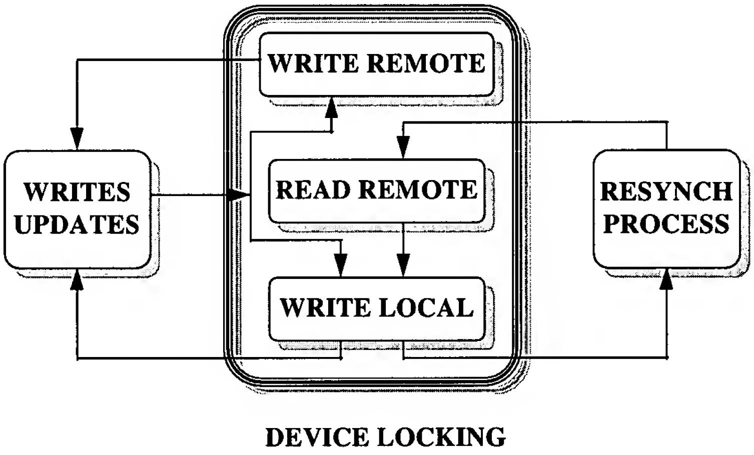


Figure 9: Coordination to recover or migrate data on-line

4. Performance Analysis

The analysis of the expected performance of the replicated disk storage system involves examining the components latency and throughput for each I/O. Since reads are not affected by the replication scheme and only occur at the primary controller, this analysis focuses on determining the performance of writes with and without replication.

As mentioned previously, cached writes complete once the data resides in memory. For the replicated system, we ensure that data resides in the memory of two controllers before posting completion. Furthermore, along with the data, we store logging information for every update which has not committed to disk.

4.1 Non-replicated System

The simple model for throughput (in I/O operations per second) for a single disk can be summarized as the inverse of the time it takes from the point an I/O is issued to the point the I/O reaches stable storage. In the case of the non-replicated storage system with cache, the maximum throughput achievable using a single disk is as follows:

$$Throughput_1 = \frac{1}{t_{TRANSFER} + t_{CACHE} + t_{OVHD} + t_{DISK}} \qquad (1)$$

where the four latency terms are defined as:

t_{CACHE} $= t_{alloc} + t_{queue} + t_{dequeue}$

t_{DISK} $= t_{latency} + t_{transfer} + t_{issue}$

t_{OVHD} = scheduling and other system overhead per disk I/O

$t_{TRANSFER}$ = data transfer time into Reflective Memory

t_{alloc} = time to allocate a cache buffer

t_{queue} = time to queue a cache buffer

$t_{dequeue}$ = time to dequeue a cache buffer

$t_{latency}$ = rotational and seek latency of disk

$t_{transfer}$ = transfer time for disk

t_{issue} = cpu time required to issue I/O

However, since the cache implementation is a write-back cache, cache hits only require a memory access. We assume that typically we will only issue a disk I/O for writes representing cache misses. We assume writes that are cache hits need only be written out to disk once. This closely models the actual behavior of the cache, since if the same data is written a number of times, consecutively within a small timeframe, the data will only be written out to disk once.

$$Throughput_1 = \frac{1}{t_{TRANSFER} + t_{CACHE} + (1-\alpha)(t_{OVHD} + t_{DISK})} \qquad (1.1)$$

In the above equation, α represents the percentage of cache hits. Note that if we assume 100% cache hits, the disk latency factor disappears from the equation altogether.

For the purpose of this analysis, we assume that all I/O is cached and discount the disk latency term. Since the Infinity is a multiprocessing system, we can proceed to determine the value of the maximum achievable throughput for cache writes as a function of the number of processors (CPUs):

$$Throughput_{\max} = \frac{1}{t_{TRANSFER} + \frac{t_{CACHE}}{CPUs}} \qquad (2)$$

$$Throughput_{\max} = \frac{CPUs}{(CPUs)(t_{TRANSFER}) + t_{CACHE}} \qquad (3)$$

Note that the cache latency is divided among the CPUs, though the actual data transfers are serialized by the data bus and therefore do not scale as processors increase.

Assuming I/O completion is reported once the data has reached the cache, then the latency for a write is simply:

$$Latency = t_{TRANSFER} + t_{CACHE} \qquad (4)$$

Given the transfer rate of the bus, we can further break down latency based on the block size of a cache request:

$$Latency = \frac{block\_size}{transfer\_rate} + t_{CACHE} \qquad (5)$$

We can thus decompose $t_{TRANSFER}$ into a linear function of the block size. The theoretical transfer rate of the Reflective Memory bus used is 100 megabytes/sec. Note that due to bus arbitration and addressing overheads, bus transfer rates, in practice, tend to be somewhat lower. A full analysis of bus performance is beyond the scope of this paper.

4.2 Replicated System

In the case of replicated I/Os, we add logging terms to the previous equations and note that now each buffer needs to be flushed to two disks. Thus, the throughput now depends on the maximum of two disk accesses.

$$Throughput_1 = \frac{1}{t_{TRANSFER} + t_{CACHE} + t_{LOGGING} + (1-\alpha)(t_{OVHD} + \max(t_{DISK1}, t_{DISK2}))} \qquad (6)$$

Assuming 100% cache I/O, we derive the equation for the maximum throughput based on number of processors:

$$Throughput_{\max} = \frac{CPUs}{(CPUs)(t_{TRANSFER} + t_{LOGGING}) + t_{CACHE}} \qquad (7)$$

where:

$t_{LOGGING}$ = t_{log} + t_{unlog}

t_{log} = time to log entry in Reflective Memory

t_{unlog} = time to clear log entry in Reflective Memory

In the case of replicated cached I/O, we can calculate the latency as previously adding a new term for logging overhead:

$$Latency = t_{TRANSFER} + t_{CACHE} + t_{\log} \qquad (8)$$

$$Latency = \frac{block\_size}{transfer\_rate} + t_{CACHE} + t_{\log} \qquad (9)$$

From this analysis, we see that latency of 100% cached writes should only increase by a constant factor (t_{log}) and maximum throughput also adds a logging factor.

However, for less than 100% cache hits assessing the overhead of replication is more complex, as an I/O synchronization factor is added which affects performance.

5. Performance Measurements

The performance evaluation of the Infinity involved running local synthetic benchmarks on the Infinity system and a DASD benchmark from a mainframe under the DASD Infinity configuration. Two Infinity systems were used for comparison in the benchmarks: a single node system, and a high availability, dual node, replicated disk system. The replicated system used the DASD dual-copy option with replicated disks residing on separate controller nodes.

5.1 Test configuration

The test configuration is equipped with 192 Megabytes of Reflective Memory dedicated to cache on each four-processor (M88100) symmetric multiprocessing controller. The high availability, dual-copy configuration uses two of these controllers while the standard configuration only uses a single controller.

It is notable that the system being tested is a prototype and has not been tuned to deliver peak I/O performance under all workloads. Thus, all performance figures presented are for the purpose of comparison only and may not reflect actual system performance.

Table 1: Response times of cache writes

Block Size in bytes	Average Response in µs non-replicated	Confidence Interval 95%	Average Response in µs replicated	Confidence Interval 95%
1024	257	±1.14	272	±2.34
2048	267	±0.49	278	±1.05
3072	283	±0.54	294	±1.96
4096	301	±0.52	313	±1.67
5120	316	±0.56	330	±1.98
6144	335	±0.52	350	±1.92
7168	348	±0.60	367	±2.57
8192	363	±0.63	380	±2.28

5.2 Synthetic Benchmarks

To assess the overhead of replicated writes, a simple synthetic benchmark was run on dual copy replicated and non-replicated Infinity systems. This test uses a single process to perform cache writes within a 100 Mbyte data set. This test was run several times before figures were generated to mitigate effects of uncached accesses.

Table 1 illustrates the response times of a single process issuing write I/Os to the cache for a replicated and non-replicated configuration. The response times represent the average of 1000 accesses for each particular block size. The 95% confidence interval was calculated using

$$95\%confidence = \overline{X} \pm 1.96\frac{\sigma}{\sqrt{n}}$$

Note that the overhead of going to a fully replicated system is constant in all cases and is approximately 5% for the smallest blocksize. Figure 11 graphically illustrates the overhead of replicated writes. It is notable that read latency remains identical in the replicated and non-replicated cases and tends to be much lower than the write latency in both cases. This difference in the latency can be explained by the synthetic benchmark, which used Reflective Memory buffers. While writes transfer data onto the reflective memory bus, reads simply return a reflective memory address containing data. Thus, there is no actual data transfer.

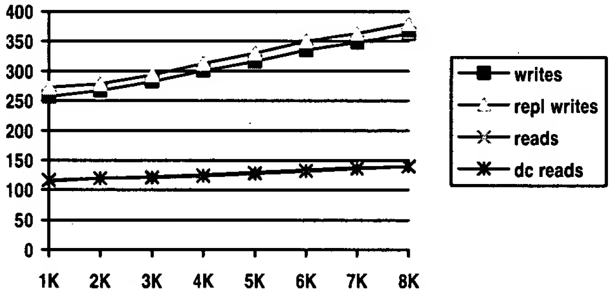


Figure 10: Response times for cached I/O in microseconds

In the graph the write latency is a linear function as predicted by equations 5 and 9 in the previous section. Using the data points for cached write latency, we can perform a least squares linear regression analysis and determine the straight line that fits the data. Table 2 indicates the linear function in the form y=mx+b. Using equations 5 and 9, from the previous section, we also determine a calculated bus transfer rate for the benchmark. The difference in transfer rates may be due to the logging in Reflective Memory which is not accounted for.

Table 2: Least squares linear regression analysis

Type	m	b	R^2	calculated transfer rate
non-replicated	15.7 μsec/kbyte	238 μsec	0.997	63.7 Mbyte/sec
replicated	16.3 μsec/kbyte	249 μsec	0.992	61.3 Mbyte/sec

In the second phase of testing, we run a benchmark that attempts to measure the aggregate throughput in I/Os per second of the replicated and non-replicated configurations. The analysis of this benchmark follows equations 3 and 7 which indicate the scalability of I/O as the number of processors increases.

The machines used have 4 processors per node. The benchmark uses 4 kilobyte cached I/O, 100% writes and calculates throughput (in I/O per second) using the time it takes for each process to execute 50,000 I/Os with multiple processes running in parallel. Figure 11 shows how the benchmark scales as the number of processors increases. In particular, note that the scalability is not linear, as predicted by equations 3 and 7, as the number of CPUs becomes large. In fact, the maximum theoretical throughput is the inverse of the bus transfer time. Using the bus transfer time calculated from the linear regression, we calculate the maximum expected throughput as the number of processors goes to infinity as $1/t_{TRANSFER}$ and indicate this reference point on the graph.

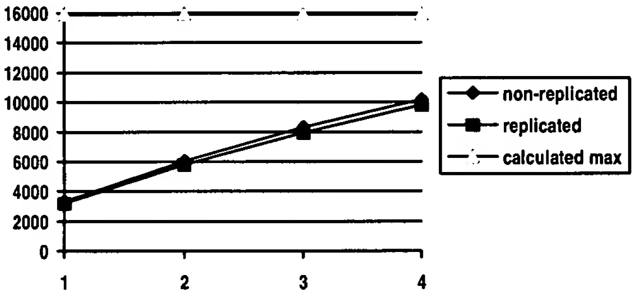


Figure 11: Number of I/O per second versus number of CPUs

5.3 DASD Benchmark

A DASD benchmark was run against the replicated and non-replicated configurations using the DASD benchmark suites from Performance Associates [3]. The benchmark was configured as a test of writes exclusively. The configuration uses 16 disks, or a total of 32 in distributed between the two nodes in the replicated configuration. The benchmark was driven by an Amdahl mainframe type 5995A using 3 mainframe disk channels to communicate with the Infinity I/O controllers.

The benchmark uses a blocksize of 4 kilobytes and issues 50% of I/O to the same logical disk track and 50% of I/O to the next logical disk track. In spite of this locality, results indicate that most accesses were to unique file blocks, generating a very small percentage of cache hits.

Tables 3 indicate the results of the benchmark. Note that the degradation for the 100% write benchmark is less than 1%. It is speculated that the added latency of data transfer from the mainframe and the fact that we are performing real disk I/O lowers the write overhead to a relatively smaller overall percentage of the transfer time.

Table 3: DASD Benchmark

I/O Type	100% writes	
Locality	50% same track, 50% next track	
Block size	4096 bytes	
Number of Disks	16	
I/O Distribution	uniform	
Configuration	non-replicated	replicated
Observed I/O rate (IO/sec)	599.9	597.1
Throughput (Mbyte/sec)	2.46	2.45
Response Time (msec)	26.65	26.76

5.4 Failure recovery

On-line recovery is not intended to be non-intrusive, but is intended to provide access to data while recovery is in progress. During the resynchronization period for a disk, average I/O response times increases since the processes issuing updates must now contend with recovery processes. The resynchronization serializes access to the disk while recovery is in progress and each pending request may have to wait for a recovery I/O (both read and write) to complete before issuing.

As demonstrated by Figure 12, in the case of cached I/O, this overhead may reach an order of magnitude, as each cached update may have to wait for a full recovery I/O consisting of physical disk read and write. However, for the period of recovery, data is still available at disk access speed rather than memory access speeds.

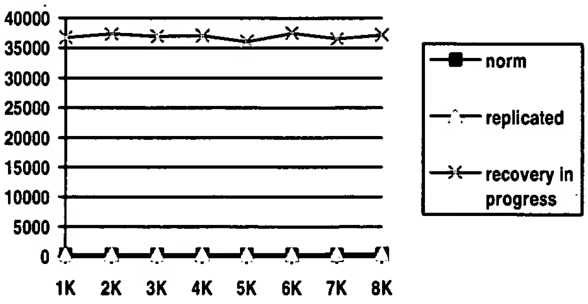


Figure 12: Response time of writes with recovery in progress in microseconds

6. Experience and Summary

The goal of the Infinity design has been to support an open, fault-tolerant architecture for massively parallel input/output services. The implementation of a replication scheme has expanded the architecture to provide high availability data storage at speeds approaching conventional non-replicated configurations.

Preliminary experience with the fault-tolerant DASD configuration has indicated that the Infinity is able to provide fault-tolerance to high performance I/O-based applications. The primary copy replication scheme enhances the system, allowing controllers to be added or removed from the configuration without affecting the behavior of end applications. Since each physical node has a separate power supply, Reflective Memory board and cabling, components can be repaired or replaced while the rest of the system remains on-line. Disks or RAIDs can be distributed or replicated across controller nodes.

This replicated implementation comprises a first step in achieving a fault-tolerant hardware and software environment for the Infinity. It is now a fault-tolerant disk storage system that survives single component failures.

Performance testing on a prototype replicated system has proven that the log-ahead protocol adds a small amount of latency to each disk or file system update. There is little added synchronization between the controllers, with each replicated update involving logging recovery data in Reflective Memory.

Failures of individual disks can also been handled on-line. Having detected a failure of a disk, operators can replace it with a new one and start an on-line synchronization utility. Because recovery is synchronized with incoming updates, updates may continue, albeit with some performance penalty.

In summary, there are three contributions which surface as a result of this work:

- Insight into the design of fault-tolerant storage systems based on massively parallel processing systems
- Implementation of low overhead replication schemes based on Reflective Memory or other distributed memories
- Utilization of a modular, scaleable multinode systems as storage systems to allow on-line recovery and resynchronization operations

7. Conclusions

Reflective Memory interconnected computers are well suited to building large scale, fault-tolerant disk storage and file systems for I/O intensive applications. The modularity of the Infinity input/output controllers combined with a log-ahead primary copy scheme, providing very low replication overhead, yield a very robust and scaleable disk storage system. Preliminary experience and benchmarks indicate that very little performance is lost in moving from the conventional Infinity storage system to a replicated system, while improvements in availability are realized by exploiting the modularity of the massively parallel architecture.

LIST OF ABBREVIATIONS

BMC:	block mux channel
CKD:	count key data
DASD:	direct access storage device
DMA:	direct memory access
FORMS:	Fiber Optic Reflective Memory Systems
LRU:	least recently used
MPP:	massively parallel processor
OLTP:	on-line transaction processing
RM/RMS:	Reflective Memory Systems
SMP:	symmetric multiprocessor

Appendix: Excerpts from DASD benchmark report

No dual-copy replication

```
1                            J E S 2  J O B  L O G  --  S Y S T E M  V R P T  --  N O D E  E N C R M V S 1
0
  18.06.37 JOB09762  $HASP373 NICKC16  STARTED - INIT    1 - CLASS A - SYS VRPT
  18.06.37 JOB09762  IEF403I NICKC16 - STARTED - TIME=18.06.37

  18.06.38 JOB09762  +I/O DRIVER LEVEL 3.0.D01 STARTING
  18.06.38 JOB09762  +NUMB-DDS 016Y03C00 A1 40 FORMAT AND DATA SET SMOKE TEST:
  18.06.38 JOB09762  +DDDEF001000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.06.38 JOB09762  +DDDEF002000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.06.38 JOB09762  +DDDEF003000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.06.38 JOB09762  +DDDEF004000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.06.38 JOB09762  +DDDEF005000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.06.38 JOB09762  +DDDEF006000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.06.38 JOB09762  +DDDEF007000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.06.38 JOB09762  +DDDEF008000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.06.38 JOB09762  +DDDEF009000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.06.38 JOB09762  +DDDEF010000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.06.38 JOB09762  +DDDEF011000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.06.38 JOB09762  +DDDEF012000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.06.38 JOB09762  +DDDEF013000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.06.38 JOB09762  +DDDEF014000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.06.38 JOB09762  +DDDEF015000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.06.38 JOB09762  +DDDEF016000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.06.38 JOB09762  +EXPERMNT01 TEST CASE # 1
  18.06.38 JOB09762  +I/O-RATE0125025025025025025025025025025025025025025025025250
  18.06.38 JOB09762  +TOTAL TARGET I/O RATE    4000
  18.09.38 JOB09762  +EXPERIMENT I/O RATE:    599.9    AVG. RESP:     26.649
  18.09.38 JOB09762  +EXPERIMENT I/O RATE:    599.9    AVG. RESP:     26.649
  18.09.39 JOB09762  IEF404I NICKC16 - ENDED - TIME=18.09.39
  18.09.39 JOB09762  $HASP395 NICKC16  ENDED
0------ JES2 JOB STATISTICS ------
-   16 DEC 1994 JOB EXECUTION DATE
-               63 CARDS READ
-              729 SYSOUT PRINT RECORDS
-                0 SYSOUT PUNCH RECORDS
-               43 SYSOUT SPOOL KBYTES
```

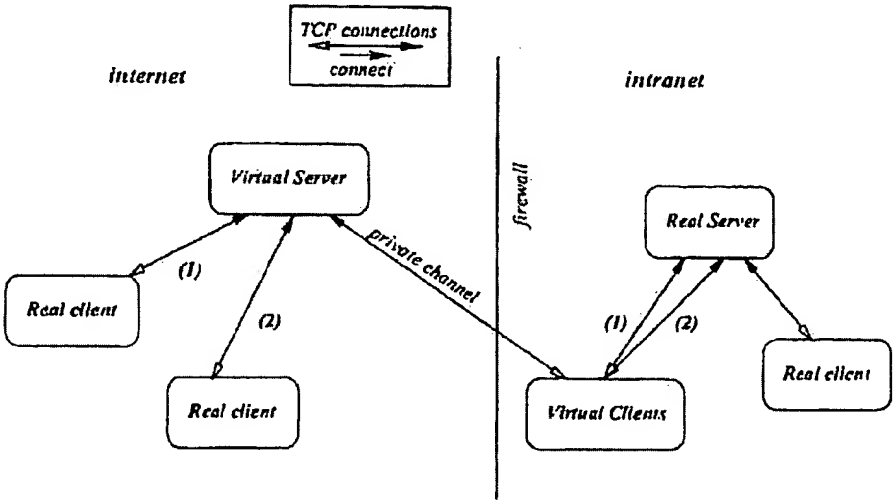
Dual-copy replication

```
1                            J E S 2  J O B  L O G  --  S Y S T E M  V R P T  --  N O D E  E N C R M V S 1
0
  18.14.36 JOB09763  $HASP373 NICKC16  STARTED - INIT    1 - CLASS A - SYS VRPT
  18.14.36 JOB09763  IEF403I NICKC16 - STARTED - TIME=18.14.36
  18.14.36 JOB09763  +I/O DRIVER LEVEL 3.0.D01 STARTING
  18.14.37 JOB09763  +NUMB-DDS 016Y03C00 A1 40 FORMAT AND DATA SET SMOKE TEST:
  18.14.37 JOB09763  +DDDEF001000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.14.37 JOB09763  +DDDEF002000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.14.37 JOB09763  +DDDEF003000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.14.37 JOB09763  +DDDEF004000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.14.37 JOB09763  +DDDEF005000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.14.37 JOB09763  +DDDEF006000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.14.37 JOB09763  +DDDEF007000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.14.37 JOB09763  +DDDEF008000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.14.37 JOB09763  +DDDEF009000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.14.37 JOB09763  +DDDEF010000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.14.37 JOB09763  +DDDEF011000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.14.37 JOB09763  +DDDEF012000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.14.37 JOB09763  +DDDEF013000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.14.37 JOB09763  +DDDEF014000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.14.37 JOB09763  +DDDEF015000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.14.37 JOB09763  +DDDEF016000100000000 50 50 00 00 00 00 00 00 00 00 10 VERY HIGH LOCALITY
  18.14.37 JOB09763  +EXPERMNT01 TEST CASE # 1
  18.14.37 JOB09763  +I/O-RATE0125025025025025025025025025025025025025025025025250
  18.14.37 JOB09763  +TOTAL TARGET I/O RATE    4000
```

```
   18.17.37 JOB09763  +EXPERIMENT I/O RATE:   597.1   AVG. RESP:    26.775
   18.17.37 JOB09763  +EXPERIMENT I/O RATE:   597.1   AVG. RESP:    26.775
   18.17.37 JOB09763  IEF404I NICKC16 - ENDED - TIME=18.17.37
   18.17.37 JOB09763  $HASP395 NICKC16  ENDED
0------ JES2 JOB STATISTICS ------
-    16 DEC 1994 JOB EXECUTION DATE
-              63 CARDS READ
-             729 SYSOUT PRINT RECORDS
-               0 SYSOUT PUNCH RECORDS
-              43 SYSOUT SPOOL KBYTES
```

AT3

Program for Exporting Transmission Control Protocol-Based Services through Firewalls



Figure

Disclosed is a program that mirrors Transmission Control Protocol (TCP)-based services and can be used to export a TCP-based service over a firewall using SOCKS v4.

The program, called tcprelay, is designed to run in either of two modes, called double-server and double-client modes, respectively. In the double-server mode, the program binds and listens for TCP connections on two ports, specified as run-time parameters in the current embodiment. In the double-client mode, the program attempts to connect to two TCP servers, also specified by run-time parameters. These modes may be thought of as "gender changers" for TCP.

Typically, the two modes are instantiated in pairs and one connection of the double-client is directed to one port, say Port1, of the double-server, as shown in the Figure. This connection will be called the private channel. The second connection of the double-client is directed to an existing TCP server, which will be hereafter called the real server. A real client now connects to the other port, say Port2, on the double-server. The program is designed such that all messages are faithfully relayed between the real client and the real server via the private channel. For completeness, for each connection made (or broken) by a real client to Port2, a corresponding connection is made (or broken) by the double-client instance to the real server, and conversely, if

© IBM Corp. 1997

the real server breaks a connection, the corresponding real client connection is broken by the double-server instance. Messages from individual real clients are multiplexed over the private channel and demultiplexed to the corresponding connections to the real server, and messages from the real server on individual connections are likewise faithfully demultiplexed to the real clients. Port2 of the double-server instance, thus, mirrors the real server, and the double-client instance correspondingly appears to the real server as a reflection of one or more real clients. The double-server and double-client modes, thus, function as virtual server and virtual client, respectively. The current embodiment allows one to limit the number of simultaneous real connections.

The program provides one way to "virtual-host" a TCP service, say an Hyper Text Transfer Protocol (HTTP) server. This by itself is sometimes useful, for example, when a web site needs to be temporarily relocated to another host without actually moving the contents. One scenario where this comes in handy is when merging multiple sites into a composite web site. More usefully, the program allows a TCP server to be "exported" to the Internet through a SOCKS v4-enabled firewall. The double-client mode is SOCKSified so that it can connect to the double-server instance running outside the firewall and establish the private channel. Real clients on the Internet can now connect to the "virtual server" outside the firewall and access the services provided by the real server inside. Real clients inside the firewall may connect directly to the real server but can optionally connect to the virtual server if they (the clients) are SOCKSified. This is particularly useful for avoiding the overhead of maintaining identical internet and intranet TCP services in sync.

The method is fairly secure so long as the real server does not provide loopholes to the real clients. If a web (HTTP) server is to be "exported", the documents and Common Gateway Interface (CGI) programs must be carefully screened for confidential content and unintended access. It is also generally possible to set up a local firewall on the host of a real server that will be exported for an extended period. Intranet links inadvertently left in the documents will also become visible to the foreign clients, but since the intranet sites remain inaccessible, the risk is actually limited to the disclosure of host names and the look and feel of broken links. The method is not limited to web servers and can be employed for exporting other TCP-based services as well. For example, using this program, one can "export" the X server running on an intranet workstation to X clients executing on a foreign host. This can be done relatively safely by restricting the number of simultaneous connections and running the virtual client instance only when actually necessary.

The advantage of the method lies primarily in its simplicity. The method avoids making changes to the firewall configuration that could have wide repercussions in case of errors and allows the exported services to be swiftly deployed, modified and relocated.

AR 4

ESG Enterprise Storage group

Storage Knowledge • Business Results

Product Brief

Product Brief - May 2000
StorageApps' SAN Appliance - SANLink

What Is It?

The SANLink SAN appliance is a storage network-enabling platform used to launch value-added storage services and applications. You can think of it as a black box where traditional host based volume management is done, along with a bevy of other add-on storage applications that are typically run on either a host or on an array itself. The SANLink represents a new way of thinking about storage architectures, one whose time has come.

Why Do We Need It?

To understand this, we simply need to understand the primary problem areas associated with storage networks, like SANs.

1. **Interoperability** - what it means is we can't easily use multiple array vendors, multiple host vendors, and multiple switch vendors within the same SAN. We want to, we just can't.
2. **Security** - the only valid way to keep NT from seeing and using our Solaris volumes within a common array is to use the array LUN mapping/masking capabilities. That's great, unless we have two or more different array types. Volume managers are great, but they differ among platforms and a universal, heterogeneous volume manager is a long way away.
3. **Applications** - sure we want to replicate our storage and we want to have instant snapshots of our data. If we run these functions in the host, they will differ among platforms. If we run it in the array, they will differ among array vendors.

Enter The SANLink

The StorageApps SANLink was designed from the ground up to mitigate all the ugly issues associated with SAN.

The Details: What Is It?

The SANLink is a black box that sits in the SAN between the host layer and the storage layer. It "virtualizes" all of the storage behind it, allowing users to use whatever storage they like as a part of a global storage pool. The product was designed from the beginning to eliminate the issues associated with heterogeneous storage and operating systems. By eliminating these issues, users are freed to always use best-of-breed storage devices as they become available, with no investment protection concerns. It's easier to think of it as now running Volume Manager off the host, in a black box.

Finally, the architecture is used to launch specific storage applications that have traditionally executed at a specific host or array. By using the SANLink appliance platform to execute these tasks, you can see the benefits immediately - I don't care about what hosts or storage I have, I run my apps against a "virtual" reality.

Applications Today

The SANLink appliance supports the following applications (part of the companies SANsuite software offering) out of the box:

1. **Data Mirroring** - synchronous or asynchronous local or wide area mirroring (a la EMC's SRDF) via either fibre channel or TCP/IP over Ethernet, GigE, ATM, T1/T3. In this way, mirroring is completely *host and storage independent*. You can mirror anything to anything. Of course, the system can be architected for complete redundancy. As an extension, the application also supports on-line data migration between LUNs.

2. **Point-In-Time Image** - this hot technology was incorporated to provide point-in-time images of data sets in its current state. There are two ways to access a point-in-time image. One is by use of an n-way mirror (ala EMC TimeFinder), and one that gives a true instantaneous image (snapshot). Either allows for on-line image functionality for backup, testing, etc. along with the ability to quickly restore from a known good state instantly, in the case of snapshot. This will aid in everything from on-line backup to new application testing, all real-time.

3. **Virtualization** - creates a logical pool of storage and allows for dynamic management of devices, which are not constrained by physical

limitations. This is completely host software independent, so any and all hosts can use/see what the appliance presents to it as LUNs. This dramatically enhances the functionality of fibre channel switches, which cannot zone to the LUN level. A side benefit is the ability to use SCSI disk right along side fibre arrays completely transparent to the host systems or underlying SAN architecture. Since the volume management/virtualization is handled in the appliance, new hosts can be added to the SAN and given access to storage on-line.

4. **Security** - The product provides secure logic masking (LUN mapping and masking) that allows for simultaneous connection by any number of heterogeneous hosts. Each host only sees what users configure it to see - with no additional host side software. Now users can easily configure NT hosts along side Unix hosts.

Summary

We expect big things from this small box. The entire SA industry has been plagued with these real-world user issu and finally someone has stepped up to address the While there are a host of other "SAN Appliance" offerin on the horizon, we think that StorageApps has the mo well thought out strategy. Several members of t management team of the company come from the end-us market - most notably, they represented senior executives at Morgan Stanley. They understand the re issues because they dealt with them daily. The compa clearly has an OEM focus right now but will deliver turnk product into a market that is screaming for this solutio They are actively growing a direct sales and supp organization in order to meet market needs. Expe consistent enhancements and new software applicatio over the next year, like Serverless backup and Enhanc Cluster capabilities. Pricing has not been discussed, b we're quite sure that it will come in less than a sing license of EMC's SRDF and offer SAN users unparallel investment protection for all their other gear.

All trademark names are property of their respective companies.

Information contained in this publication has been obtained by resources the Enterprise Storage Group (ESG) considers to be reliable but is not warranted by ESG. This publication may contain opinions of ESG, which are subject to change from time to time.

This publication is copyrighted by the Enterprise Storage Group, Inc. and is intended only for use by subscribers. Any reproduction or redistribution of this publication, in whole or in part, whether in hard-copy format, electronically, or otherwise to persons not authorized to receive it, without the express consent of the Enterprise Storage Group, Inc., is in violation of U.S. copyright law and will be subject to an action for civil damages and, if applicable, criminal prosecution. Please contact us at (508) 482.0188.

AS4

Enterprise Storage Group

Enterprise Storage Report Technical Review

volume 6 | published monthly by the Enterprise Storage Group
www.enterprisestoragegroup.com

Early adopters of Storage Area Networking (SAN) technology will find that SANs do not yet live up to the promise of simplistic universal storage. Traditionally, the problems associated with secure access control have had few remedies, which only offered limited help. The result has been slow adoption of SAN technology overall.

This report discusses the advent and definition of a SAN appliance, which, simply stated, makes SAN technology usable. It also contains a review of leading manufacturers' SAN appliance products - what's available today, what to look for in the coming months.

In researching this report, we received input from over 200 end users and spoke with leading industry vendors including Veritas, Compaq,

San Appliances

conclusion

SAN technology has not lived up to the glorious end all that was promised to the market years ago. The predominant characteristic that causes users the biggest problem is its inability to share heterogeneous components (multiple array types, switch types and operating systems) in a single, unified SAN. The SAN appliance, while too long in coming, has finally arrived to help users solve this problem. The two companies to watch early on are StorageApps and DataCore. However, others could make a move into the market during these early stages.

What a SAN Should Be

The point of SAN was to enable users to consolidate disk resources in a virtual fashion. Once all the disks are consolidated into a "pool" of storage resources, carving up the pool into whatever logical devices desired allows access to those devices by selected hosts. All of the hosts, regardless of OS, can be connected via high-speed switched busses to the storage pool. The result is a secure environment whereby specific hosts only have access to selected logical volumes.

Factoids

- *83% of user respondents claimed interoperability among various components as the most significant problem with SAN technology.*
- *77% of SANs are implemented as homogeneous SAN "islands" (one type of OS, one type of switch and one type of storage array).*

The Realities of SAN

Little of what was just described above exists to date. Enterprise Storage Report Technical Review, Volume 1, Storage Area Networks, details the problems associated with secure access control (LUN masking and mapping) - the ability to have a SAN host see and access only certain logical or physical volumes. This is still a problem. Traditionally, there were only a few ways to address this issue.

Switch zoning. This method works well if zoning an entire array, but is effectively useless since it cannot zone to the LUN level.

Array control. The array does the "volume management" by creating logical devices and presents them to specific hosts. This is the only real alternative to date, but the downside is that it only works on that specific array. Since users have more than one array type in their shops, they usually don't want to be restricted to using any one vendor.

Host software control. SANergy (IBM) or Transoft (HP) is good at back-end disk virtualization, but offers limited heterogeneity at the host OS support level. We do not agree with a long-term strategy that requires host drivers/software because of increasing complexities as things are added over time. Driver writing/maintaining is tedious.

What is a SAN Appliance?

At a fundamental level, you can think of it as a universal, global (to the SAN) Volume Manager in a box, or a series of connected boxes. The appliance runs software that "virtualizes" all of the storage behind it in the SAN, and presents selective volumes (typically as LUNs) to specific hosts. It also precludes other hosts from seeing or accessing those LUNs. More advanced appliances will offer an extensible platform from which to launch additional storage applications, such as multi-mirrors, snapshot, replication, serverless backup, etc. The SAN appliance is the perfect place to not only collect data and virtualize storage pools, but also to launch these storage centric applications - providing users a single platform to manage storage applications regardless of storage type or server type. Although we are not there yet, there is a flurry of activity to fill huge demand in the user base for such a functionality. In a perfect world, the switch vendors would have considered this basic SAN "usability" issue and incorporated the features long ago. If they had, SAN adoption rates would most likely be light years ahead of where they are currently. More likely than not, switch vendors will wholeheartedly support the efforts of the appliance folks since they stand to benefit more than all others. In other words, as soon as SAN is more usable, users will start eating up switches.

Why We Need SAN Appliances

Simply stated, SAN appliances make SAN technology usable. All of the SAN appliances discussed herein present virtual LUNs, and prevent those LUNs from being accessed by rogue hosts/servers. They effectively eliminate most OS interoperability issues between vendors and offer a single point of management. More importantly, however, is the fundamental premise that there is now a unified platform from which to develop and launch storage applications outside of both the host/servers and storage. This is important because users are now truly free to choose the best technologies without restriction or fear that a new product that replaces an expensive component you just purchased will be available the very next month. If EMC is the best array for you today and you purchased HDS last year, SAN appliance allows you to run them side-by-side. Finally, don't fret if and when NT creeps into your Solaris SAN. You can mask the NT server from seeing any of the Solaris storage, and vice versa, through the appliance. Any user who has attempted to put any sort of heterogeneous SAN together has experienced this frustration. SAN is difficult enough in a homogeneous environment, let alone a mixed bag. The true promise of SAN, however, is only realized when homogeneous restrictions are removed and the SAN is used properly - as a high speed interconnect of heterogeneous hosts, connected to a common pool of disparate disk types, all virtualized and logically connected in a secure, simple fashion.

The market for these SAN appliances is brand new, and none of today's solutions are perfect. All potentially will get better. Suffice it to say, however, that you really cannot make a bad choice here, since all offer better hope than what was previously available.

Terms To Know

SAN Appliance - a dedicated box(es) that performs storage virtualization and management within a SAN.

In-Band - an appliance that sits in the data path, between hosts and storage. In-band means host software/drivers are not required, so ease of use in enhanced. Also, more data can theoretically be gathered without requiring host cycles. Being in the data path is a problem, if there is a data path problem. In other words, you are only as fast as your slowest component (today that is disk). So, in theory, you could max out a poorly designed In-band architecture and cause overall performance problems, although we don't foresee this being a problem any time soon.

Out-Of-Band - an appliance that is in the SAN, but not in the data path. Out-of-band appliances will require drivers at each host connected to the SAN. The positive side to this approach is, effectively, infinite scalability since the appliance does not consume bandwidth. The negative side is the amount of data that can be gathered varies by host and requires lots of different drivers and extensive maintenance. There will also be some minimal performance issue on the host.

The SAN Appliance Players

Compaq

Compaq *(www.compaq.com)* is the first system vendor to validate the need for the appliance, which isn't surprising since they are shipping more pre-canned SAN solutions than anybody else. Compaq recently announced their intentions for supplying an appliance in sparse terms - in other words, it isn't ready for prime time, doesn't really do anything yet, but they understand the need in the marketplace. The appliance is an out-of-band approach that will host some applications, which resulted in an OEM agreement with HighGround for SRM (storage resource management). The product holds promise, since Compaq is a proven volume mover of SAN product and is seriously committed to the storage sector. Keep an eye on their progress over the next six months. They are expected to have a full function solution, albeit only for their own storage at first.

StoreAge

StoreAge *(www.store-age.com)*, the Israeli-based start-up, has brought some outstanding functionality to the market right up front. Their SAN appliance performs out-of-band SAN management and provides a relatively complete list of essential functions.

The appliance provides:

- A centralized view of the SAN, even with multiple appliances (which can be configured for fault tolerance) via a common web interface.
- Array pooling, allowing virtualization of heterogeneous arrays.
- SAN scalability because there are no pre-defined limits on SAN growth since it's not in the data path.
- Security - selective presentation of LUNs to each host. Standard fibre channel switch zones are created between host and appliance (the hosts never see the actual physical storage).
- High availability. Complete redundancy is supported including appliances through multiple host bus adapters (HBAs).
- Cluster support. LUNs can be presented to multiple hosts for cluster configurations.
- Interoperability. Multiple heterogeneous storage arrays can comprise a single pool. Multiple heterogeneous hosts see only what you allow them to see.
- Performance. StoreAge provides the ability to stripe across physical arrays and switches. Dynamic load balancing between HBAs is automatically provided. The StoreAge driver allows effectively unlimited bandwidth scalability within each host. The differentiator being that if more I/O out of the HP/Oracle node is needed, inexpensive HBAs can be added to the systems, which automatically aggregates the additional available throughput and I/O.

The potential pitfalls to StoreAge, beside their relative newness, are several fold. StoreAge is a very small company without a major OEM supporter to date. Without a major OEM, the company most likely will not be able to reach a broad based market on their own. The only technical negative is the out-of-band issue. The company is small and may not have the ability to support all the myriads of drivers/OS that are out in the constantly changing marketplace. Aside from those issues, look forward to more creative thinking from the likes of StoreAge.

DataCore Software

DataCore Software's *(www.datacoresoftware.com)* SANsymphony appliance software requires users to either configure their own hardware, deal with an OEM such as Gadzoox or TrueSAN, or find a local reseller to configure the hardware and software. At a fundamental level, DataCore is on target in terms of providing product with the needed core functionality. The final turnkey solution is an in-band appliance that not only virtualizes back end storage (and presents logical volumes to secure host connections), but also provides a methodology to launch storage specific applications. The product supports snapshot and instant image today, and soon will support synchronous remote copy (via fibre channel). Asynchronous support is planned by year-end, over IP.

Today, the SANsymphony software sits on an NT 4.0 platform (capability will expand to W2K this summer). The use of cache is DataCore's primary differential. Cache is used for both read ahead and write operations. Users should only configure dual-redundant appliances if write cache is used (mirror via host based volume managers). It remains to be seen whether cache will boom or bust. There will not be enough available data until there are more installations. Regardless, DataCore is well ahead of most in the field.

The only potential negative is, as a software only company, the inevitable competition with Veritas (discussed later in this review), a company that clearly has lots of volume manager, backup, file system and cluster technology to draw from.

)ataDirect Networks, Inc.

)ataDirect Networks' (DDN) (*www.datadirectnet.com*) ;AN DataDirector is an in-band black box designed to)rovide volume management as well as act as the RAID ɜngine for the SAN, so that users can plug inexpensive IBOD drives directly into the appliance.This may seem like a good idea if you have JBOD disk hanging around, but it isn't.While the appliance is good, the RAID concept if fundamentally flawed. DDN claims to have the fastest RAID engine around, which may be true.The purpose, however, of the SAN appliance is to make life easier, and managing, cabling, and dealing with failure, for example, of a multitude of individual disk drives will not make anyone's life easier.

From the core appliance perspective, the DDN box is somewhat the "director" of appliances. It is a modular, high-end package (which, like a director, is too expensive) that does all the required fundamentals. It seems as if DDN spent too much time thinking about RAID and not enough time thinking about usable storage applications, at least in the early going.

StorageApps

StorageApps (*www.storageapps.com*), while new like most, is the pick of the litter.The product, which ships in volume next month by a major OEM who we can't mention yet, is a turnkey, in-band appliance that not only provides necessary volume management functions, but also provides an infrastructure for storage application launching including replication (remote mirroring) from and to anything, and instant image (either snapshot or third mirror). StorageApps has already recognized the coming convergence of storage and IP networking and has provided for it by including the ability to run gigabit Ethernet right inside the box. Perhaps the best news is that users reap the benefits of virtualization up front since the product supports remote mirroring, replication, etc.That means not only can users have different array types locally, but they can also remotely run replication between dissimilar arrays. Users often have more expensive EMC on the production side, but cannot justify the expense on the remote side. Now, they don't have to. Users can simply use inexpensive disk and replicate between the two.With this approach, any storage can be made useful.

Not only does StorageApps have a six-month lead above most of the others, they also think like users. One unfair advantage the company has is that they recently just were actual users - a large number of senior IT management from Morgan Stanley joined forces with RAID vendor RAID Power to form a company with a unique perspective. It's tough to argue with folks who built a product needed to run a multi-billion dollar worldwide IT operation.That insight is evident in the appliance design.

The SAN appliance is really an enabling platform from which users will launch storage specific applications without host or storage array intervention. StorageApps already has landed a huge OEM, and even bigger things are expected from this group.

Veritas

Veritas Software (*www.veritas.com*) is an automatic contender whenever the words Volume Manager are issued. The company is in process of re-packaging and re-launching its SAN appliance suite (Storage Appliance) that runs on Solaris systems to create a turnkey appliance.There are actually two variants of the software, one for SAN (discussed here) and another for NAS (Veritas bundled their file system, volume manager, quick log, and a supported version of SAMBA for CIFS to allow Sparc users to create a robust NAS device out of standard hardware).

The SAN version, or block level appliance, offers volume management virtualization, but not much else. It is similar to DataCore's software approach. But, if Veritas gets serious, bet your money on them ultimately.Veritas is slightly behind some of the others that offer turnkey solutions as far as application support is concerned, but not for long. The reason they have an advantage over DataCore could also propel them up the food chain quickly, namely,Veritas has a lot of software. Clustering, backup and replication are already part of Veritas' suite of products, so incorporating them will be an obvious win.Today, the product is rudimentary (no HA support yet, so it does present a single point of failure currently).Veritas is not a company to take lightly, especially so close to home.

Veritas' only problem is that they own the volume manager space. For the appliance concept to be successful,Veritas would have to cannibalize their existing success to a large degree. It is unclear whether or not they will do that. Also, if DataCore can get out ahead of Veritas in volume shipments, users may be hard pressed to switch from a proven platform.

Vicom

Vicom (*www.vicom.com*) is arguably the granddaddy of the SAN appliance space.With more than 4,000 claimed units shipped (mostly into the IBM SSA space), they are the industry's best-kept secret.Their involvement in appliance space is surprising.At a base level,Vicom offers bus-to-bus routers (Fibre Channel, SSA, SCSI, Ethernet) that enable SAN virtualization (there is a single management view of multiple Vicom boxes). One key differentiator is they are not monolithic in the sense that their function does not come from a single black box, but rather from a modular box approach whereby users can place a Vicom box in the data path of each host (which allows you to mitigate heterogeneous OS issues, as well as to connect non-fibre channel hosts to the SAN) and on the back of the switch(es) to virtualize storage. Since Vicom was added late to this report, user feedback was unavailable. However, Vicom's number of installations and low cost merit consideration.

ConvergeNet

Dell spent a lot of time and money acquiring ConvergeNet, which was supposed to deliver the best SAN appliances. To date, they have been plagued with technical problems and a mass exodus of people. It would be surprising if ConvergeNet delivered any working technology. Dell soon is expected, on the other hand, to launch a different working initiative.

Summary - Why You Need a SAN Appliance

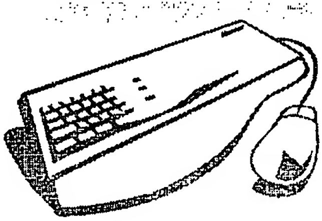
None of the products we discussed will hurt you, rather, they will only help. If you run a vanilla shop with only one OS and disk array type then you most likely will get away without using a SAN appliance, unless you use EMC storage and don't want to raise additional funding to pay for their cool add-ons like SRDF or TimeFinder. If you plan to grow your SAN to incorporate any heterogeneous elements, you need a SAN appliance.

XSP space will benefit from this technology. Not only will they be able to carve up secure pieces of infrastructure, getting better economies of scale, but also have a universal application platform that includes accounting functions.

All in all, the SAN appliance represents one of the most useful tools developed since the tape drive. The possibilities, once a new infrastructure is enabled, are endless. The result is serverless backup, enhanced clustering, SCSI over IP -- things that are happening best on a SAN appliance.

Next Month:

Storage Over IP



Enterprise Storage Group, Inc.
189 West Street, Suite 200
Milford, MA 01757
Phone: (508) 482-0188
Fax: (760) 874-8816

All trademark names are property of their respective companies.

Information contained in this publication has been obtained by sources the Enterprise Storage Group (ESG) considers to be reliable but is not warranted by ESG. This publication may contain opinions of ESG, which are subject to change from time to time.

This publication is copyrighted by the Enterprise Storage Group, Inc. and is intended only for use by subscribers. Any reproduction or redistribution of this publication, in whole or in part, whether in hard-copy format, electronically, or otherwise to persons not authorized to receive it, without the express consent of the Enterprise Storage Group, Inc., is in violation of U.S copyright law and will be subject to an action for civil damages and, if appropriate, criminal prosecution. Please contact us at (508) 482-0188.

Enterprise Storage Group

Enterprise Storage Report
Technical Review

AT4

factiva

Dow Jones & Reuters

Data Communications

NETWORK DESIGN

BUILDING a Storage-Area Network SANs boost performance, reliability, and scalability of the critical link between servers and storage devices

BY BRENDA CHRISTENSEN, Brocade Communications Systems Inc.

2,389 words
21 April 1998
Data Communications
67
Vol. 27, No. 6
English

(Copyright 1998 McGraw-Hill, Inc.)

WANT TO TURN corporate data into gold? Better keep it moving--and fast. But net managers in a rush to get the lead out often overlook a critical bottleneck: the link between servers and storage devices. n The problem is that most of those connections still rely on SCSI (small computer systems interface). SCSI is a point-to-point technology. It doesn't scale well. And while it sufficed for 1980s-style storage needs, it's coming up short today--since processing power and storage requirements have grown exponentially in the decade that's passed.

Maybe it's time for a SAN (storage-area network). SANs are gigabit-rate networks that rely on Fibre Channel for higher throughput, greater distances, and more connectivity options between servers and storage devices. They can be built as switched- or shared-access networks, and either way they offer better scalability, fault recovery, and diagnostic information than current approaches. What's more, there's no need for a forklift upgrade. And the initial outlay for SAN infrastructure could pay for itself in terms of reduced management costs over time.

When does a SAN make sense? Whenever data integrity and availability are stringent requirements. Small workgroups may benefit if there's lots of mission-critical data involved. And SANs can help larger organizations as well; here, the challenge is defining the business objectives and matching them to SAN design possibilities.

SAN Specifics Getting straight on SANs means looking at some of the basics. They generally comprise servers (hosts), storage devices (tapes and disk arrays), and bridges and multiplexers, all connected to Fibre Channel switches. As with LANs or WANs, the switches furnish the backbone for all connected devices, with one or more switches acting as a Fibre Channel switching fabric. SAN switch fabrics allow attachments of thousands of nodes.

SANs also can incorporate Fibre Channel arbitrated loop (FC-AL), a type of shared-media network. The FC-AL architecture allows for up to 126 devices per loop, either attached directly to Fibre Channel switches or to hubs that in turn connect to switches (see ``Launching a Storage-Area Net,'' March 21, 1998; http://www.data.com/roundups/san).

Further, Fibre Channel SANs help take the load off servers, which have historically been given the added burden of transferring data to storage devices and to the LAN. Now, servers can offload data transfer to the SAN--and return to their original processing role.

SAN Setup As for setting up the SAN, the job is relatively easy. Fibre Channel SANs can be designed as both shared-media and switched-access networks. In shared-media SANs, all devices share the same gigabit loop. Trouble is, as more devices are added, throughput goes down.

While this might be acceptable for very small environments, a backbone based on Fibre Channel switches will increase a SAN's aggregate throughput. One or more switches can be used to create the Fibre Channel switching fabric. Accessing the services available from the switching fabric is possible only if the NIC (network interface card) of each storage device can connect to the fabric as well as to the operating system and the applications. Basically, the NIC becomes a citizen of the network by logging into the fabric. This function is simply called fabric login, and it's obviously important to use NICs that support fabric login in building a SAN.

Another key issue for devices attached to the.SAN is the ability to discover all the devices in the switching fabric. Fibre Channel defines a discovery mechanism--SNS (simple name service)--that learns the address, type, and symbolic name of each device in the switching fabric. SNS information resides in Fibre Channel switches, and NICs and storage controllers request SNS data from the switches. Thus, net managers should look for Fibre Channel NICs and storage controllers that back SNS.

For error recovery and fault isolation, Fibre Channel has an optional feature called RSCN (registered state change notification), which issues updates to devices on configuration changes. RSCN makes the most sense when RAID (redundant array of independent disks) arrays, so-called JBODs (just a bunch of disks, a disk collection without a RAID controller), tape devices, and hosts are directly attached to a switching fabric rather than to shared-access loops, since faulty nodes won't affect any other fabric-attached devices. Switched networks also recover from faults a lot faster than shared-media nets because problem devices or links can be isolated.

Fibre Channel also has a built-in multicast feature. The alias server, an optional switching fabric service, acts as the Fibre Channel multicast clearinghouse, allowing creation and removal of multicast groups. But Fibre Channel multicast differs from IP multicast, where hosts set up multicast groups that work only at the network layer. Instead, the switching fabric aids multicasting; membership is based on the Fibre Channel physical address and is transparent to the upper-layer protocols.

Because Fibre Channel can integrate diverse protocols, net managers can build SANs that accommodate even the largest data center's storage needs. For example, it's possible to incorporate Escon (Enterprise Systems Connection) and SSA (serial storage architecture) devices into a SAN using SNA-to-Fibre Channel gateways and Fibre Channel-to-SSA bridges. Future mainframes and SSA devices could support direct Fibre Channel attachment. Using distributed lock manager software or some form of zoning mechanism, these data center SANs will permit storage to be shared among all servers in environments running a mix of proprietary operating systems, Unix, and NT.

Managing a SAN As for management, networkers should be able to use all the tools and systems they use for LANs and WANs. That means they should look for SAN devices that can be managed via SNMP or through the Web, via HTTP (hypertext transfer protocol). Devices also should support telnet (for remote diagnostics or servicing). Another option for reporting on SCSI devices is SES (SCSI enclosure services).

All of these should furnish detailed information on device status, performance levels, configuration and topology changes, and historical data. Key status and performance information would include throughput and latency metrics. Future requirements will include a commitment to delivering tuning and optimization tools.

In FC-AL networks, the hub furnishes management information on all devices within the loop. But the hub can't report on devices outside the loop. Of course, when loops are attached to a switching fabric, remote management and diagnostics are possible for all devices.

When deciding on what scheme to use for connecting loops to switches, net managers should look for features that make the best use of loop tenancy--the time a device exclusively occupies the loop for data transfer. Consider devices that collate packets intended for a single target, rather than adding the overhead of arbitrating for loop tenancy for each packet to be delivered.

The Right Mix? Up until now, net managers have installed either switched or shared-media SANs. But companies are now beginning to combine the two. The tricky part is figuring which parts of the SAN to deploy as switched and which as shared.

For small sites (one or two servers and one or two RAID arrays or JBODs) there are three choices: stick with a SCSI-only model; move to a point-to-point Fibre Channel network (with one NIC connected to each storage device); or implement a shared-access (hub-based) SAN using a loop topology. The main considerations are cost and ability to grow.

Networks with multiple servers and intelligent arrays housing 500 Gbytes or more of data are candidates for immediate adoption of a switch-based SAN. Consider a midsize company with three servers, multiple RAID arrays, and a tape system: A mix of switched attachments to RAID arrays, servers with arbitrated loops for multiple JBODs, and a bridge-attached tape system may be appropriate (see Figure 1). And remember: Because this SAN is built around a switch, it allows for fault isolation and other switched fabric services.

Factor the Factors But how much of a SAN should be shared and how much switched has to be determined on a case-by-case basis. Don't get caught up in the idea that the amount of data stored is the main factor in deciding what type of SAN to build. Instead, weigh just how mission-critical the data is; the distance requirements; the management of storage devices; the availability and disaster-recovery requirements; and the ability to manage or cope with configuration changes.

Start with how critical data access is to an organization. In storage setups with parallel SCSI links, for example, the server is the control point. When a server fails, it may take 30 to 90 seconds for a reset. That could add up to thousands of dollars for a company servicing online billing transactions. So don't opt for a shared-media SAN, since it doesn't eliminate the reset time; arbitrated loops go through a loop initialization process (LIP), causing the reset of all devices. If access to data is a competitive requirement, switched fabrics are the way to go.

If storage requirements include any real distances (such as across a building or multiple buildings in a campus), SCSI probably won't be suitable. That's because there's a length limitation of about 70 meters, even with SCSI hubs and repeaters.

As for monitoring the status of devices in multiple buildings, Fibre Channel SANs offer built-in management facilities. Departments using JBODs on loops can be connected to switches on SAN backbones. Switched backbones then create a virtual data center by directly attaching servers and arrays with the loops, giving net managers the management data.

Finally, when it comes to disaster tolerance, a switched fabric is the right choice. Creating redundant data centers 10 kilometers or more apart requires high bandwidth synchronization that only switching offers.

Medium Fit To see how SAN concepts can be put to work, consider a medium-sized company looking to upgrade its storage capabilities. Its campus consists of four buildings, two of which contain two Windows NT servers each, and two of which house four Unix server farms each. All the servers uses redundant RAID arrays connected via parallel SCSI links.

One of the first things the company should address is the sharing of I/O and file systems. Windows NT is a so-called shared-nothing environment relative to I/O and file systems, in that multiple NT servers can't share these resources. Therefore, net managers will need to implement a third-party distributed lock manager to share files and storage. Many versions of Unix have distributed lock capabilities for I/O and file systems. With these capabilities in place, it's then possible to begin building the SAN.

As for the switched/shared decision, management requirements may tip the balance toward a switched SAN in this case. With multiple sites and servers, troubleshooting is difficult. The relatively small cost difference between a shared design with arbitrated loops and a switched fabric is more than compensated for by the management information available from the switched setup. A switched fabric is also a better choice if high availability is a requirement. Using distributed lock management or a zoning mechanism, net managers can design the SAN to isolate workgroups (like those in marketing or engineering) or operating system environments (like NT and Unix), all while using the same switch.

For backing up the storage, this network design can include tape devices attached directly to the switch fabric. Net managers can then set up temporary time-of-day zones to do backup for both Unix and NT environments.

For disaster tolerance, this design would benefit by placing the backup system 10 km away from the switched SAN. Additionally, the redundant RAID devices can be separated from the server farms and instead directly attached to a Fibre Channel switch. This allows RAID arrays to be located in different buildings, further enhancing redundancy.

Moving Mainframes to SANs When it comes to mainframe data centers, using a Fibre Channel switching fabric pays off in terms of higher scalability and lower cost of ownership. Escon-to-Fibre Channel bridges can't increase the performance of an Escon director (which is analogous to a switch in Escon terminology), but they will increase the longevity of legacy gear and help achieve many-to-many connectivity.

With Escon, there's always a path available to allow for continuous computing in the event of node failure. Shared-media designs are thus not appropriate. Escon directors can't connect more than two other director units, and one becomes a fan-out device for shared-access attachments. Therefore, scalability beyond the second director is not possible.

Fibre Channel switches, in contrast, can be intelligently cascaded. Because of their any-to-any connectivity, Fibre Channel SANs allow automatic path selection. Cascaded switches also distribute management software among all devices. If one switch fails, management information is available to all fabric devices not directly attached to that switch.

There are other benefits--like the ability to increase port count. Currently, Escon directors can support a maximum of 256 ports--but by moving mainframe storage connections to SANs using cascaded Fibre Channel switches, that can be boosted to 400 or 500 ports. And when used in conjunction with shared-access loops, the number of attached storage devices can climb into the thousands.

Today, disaster recovery and resource planning require net managers to maintain the reliability and availability features of the glass house but in a distributed manner. To this end they can use SANs to build data centers 10 to 30 km apart. One caveat with this design, however: To adjust for the longer distances, the time-out values of login processes for all attached devices have to be increased.

Cascading switches also makes management easier, since there's a single view of the switching fabric. Net managers in the network operations center can thus keep an eye on the metropolitan-area SAN.

BRENDA CHRISTENSEN is vice president of marketing for Brocade Communications Systems Inc. (San Jose, Calif.). Her e-mail address is brenda@brocade.com.

Illustration: Switching and Sharing

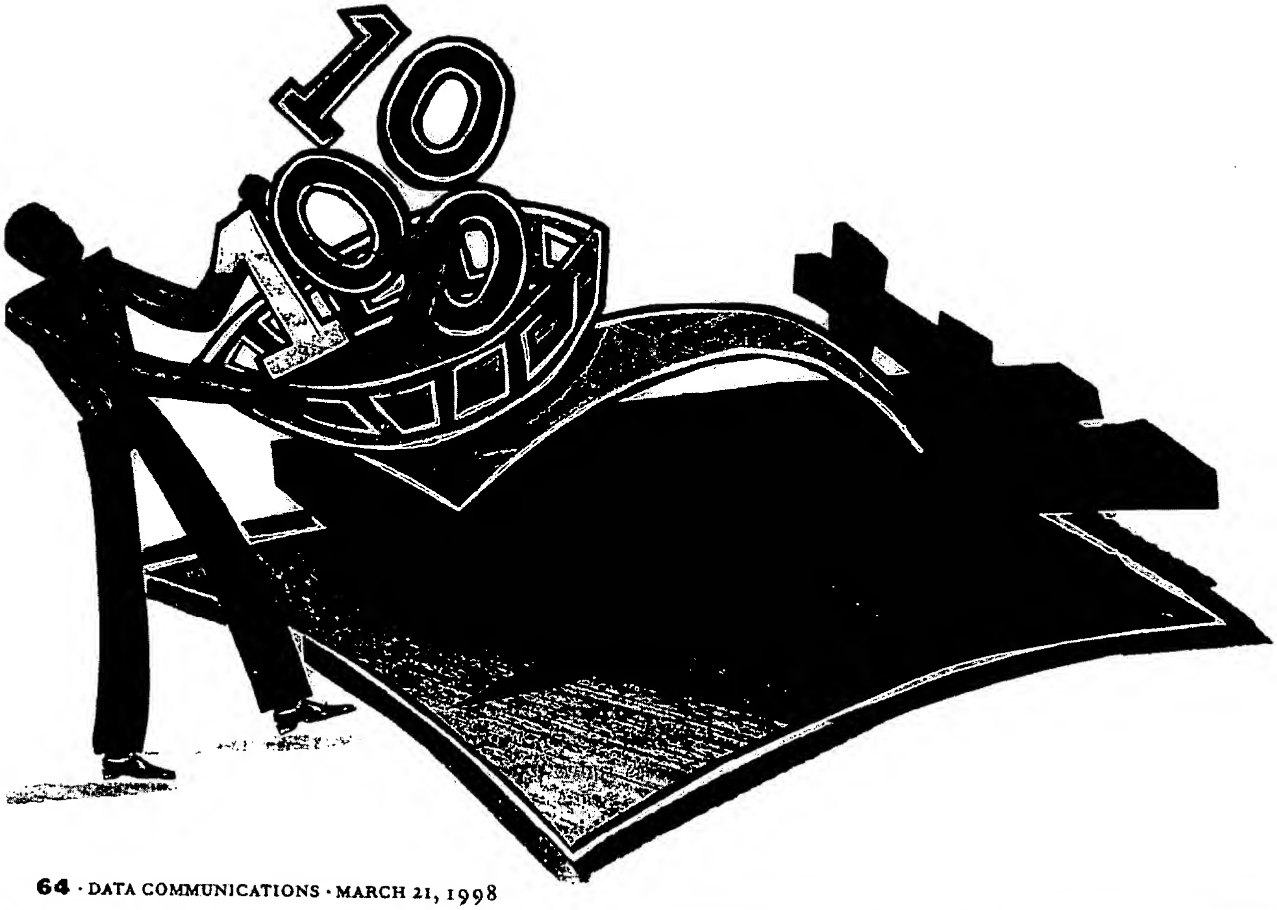
Document dcom000020010919du4l000ad

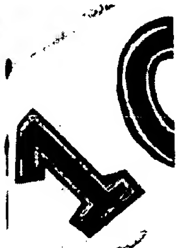
© 2005 Dow Jones Reuters Business Interactive LLC (trading as Factiva). All rights reserved.

AR 5

Launching A Storage-Area Net

High-speed storage-area networks help net managers hurl stored data around quickly—and take the load off the LAN and WAN **BY MARY JANDER**





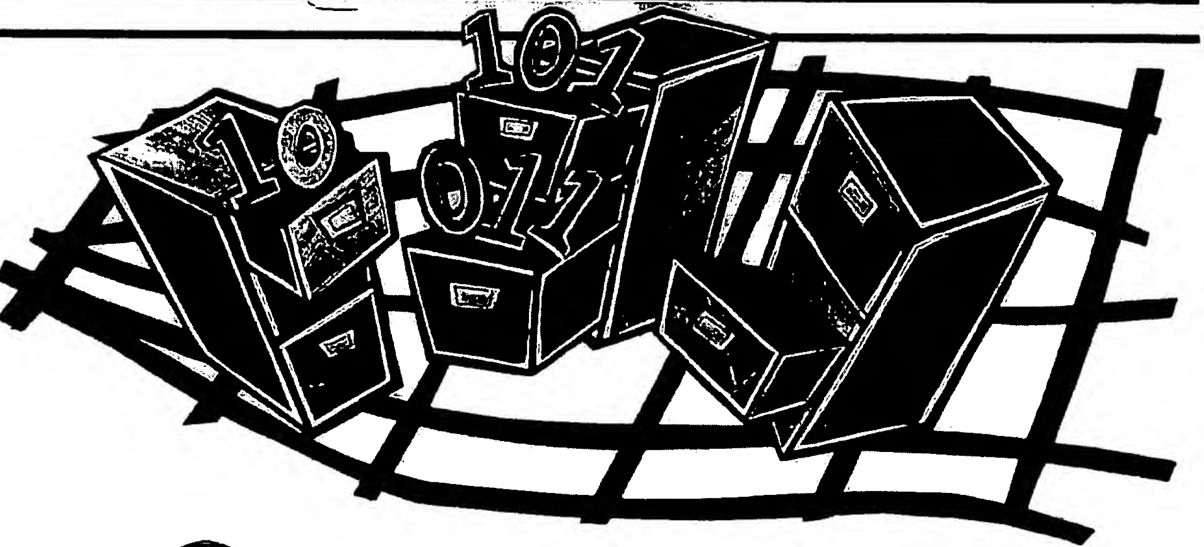
O

see now. Clie
cellany of pr
Internet have
strain on the
between host
get about mo
tune with th
dealing with
make for a lo

Maybe i
storage camp
(storage-area
cated, paralle
for backing u
corporate ne
Gbit/s fibre c
need to worry
huge amounts
ing a separate
than ease the
also helps ke

MARY JANDE
*editor/special p
e-mail address i*

ILLUSTRATION



Out of sight, out of mind? If that's the philosophy on storage, net managers who haven't taken a look lately might not like what they see now. Client-server networks, a miscellany of protocols, and the rise of the Internet have combined to put a serious strain on those old-school SCSI links between hosts and storage devices. Forget about monolithic arrays humming in tune with the mainframe: These days, dealing with loads of stored data can make for a load of trouble.

Maybe it's time to launch a new storage campaign—and set up a SAN (storage-area network). SANs are dedicated, parallel networks built especially for backing up and archiving the data on corporate nets. They're based on 1-Gbit/s fibre channel links, so there's no need to worry about speed when tossing huge amounts of data around. And having a separate net for storage does more than ease the warehousing hassles; it also helps keep massive storage transfers from clogging the pipes on the main corporate network. What's it all add up to? Just "the most significant development in storage we've seen in 15 years," according to Michael Peterson, president of Strategic Research Corp. (Santa Barbara, Calif.), a consultancy that specializes in storage management.

Mary Jander *is associate managing editor/special projects for* Data Comm. *Her e-mail address is mjander@data.com.*

But make no mistake: Rewriting the story on storage means constructing an entire network from scratch—beginning with the topology. SANs can be built with switches or hubs as the key connectivity devices, and each presents its own set of concerns. Fibre channel switches may offer dedicated 1-Gbit/s links, but products (and standards) are in short supply. Shared-media hubs split that 1 Gbit/s among as many as 126 nodes, but the gear is here, the standards are well defined, and the prices are a lot lower. If switches are chosen, check that they can handle the right mix of service classes and find out how the backplane is set up. And with both switches and hubs, look into capacity, redundancy, and management. After that, consider cabling: Fibre channel distance limitations differ depending on whether copper or fiber is used. Then take a look at the gear used for linking the SAN to the LAN—and keep in mind that getting it all to work together isn't a given. So do some hands-on testing before simply hooking fibre channel connections into legacy equipment.

Still, net managers who follow these SAN steps not only will keep their sanity—they could also reap big rewards. Ask David Grandin, president of Avid Sports LLC (Lowell, Mass.), which builds video networks that professional sports teams use to review their plays and performance. He says SAN technology permits simultaneous access to videos stored on multiple disk arrays, something LAN technology doesn't. That's won Avid more customers—and more profits.

Sanity Check

Not quite sold on SANs? That's OK: Building an entirely new network—and taking on the configuration, management, and maintenance chores that go with it—can be daunting. Net managers are right to wonder whether it's worth the hassle.

Building an entirely new network may seem daunting, but consider the alternatives.

But consider the current alternatives. There are two other choices for connecting storage devices to the network. One is to hook RAID arrays, tape storage libraries, and optical jukeboxes directly into the LAN. The other is to use the SCSI (small computer systems interface) standard to funnel data back and forth between hosts and servers. The problem with the first? It kills LAN per-

ILLUSTRATIONS BY TOM GARRETT

formance. The problem with the second? SCSI doesn't work beyond 25 meters with copper, and it doesn't work at all with fiber.

Now consider a SAN. It's based predominantly on 1-Gbit/s fibre channel connections and has its own switches, hubs, and gateways. These are linked to the servers and hosts on the corporate intranet, which furnish the point of contact between it and the LAN (see Figure 1). And because fibre channel doesn't suffer the distance limitations of SCSI, the SAN can be extended over the entire site—delivering 1-Gbit/s rates at up to 30 meters over copper, half a kilometer over multimode fiber, and 10 kilometers over single-mode fiber.

What's more, fibre channel frames have a lot more room for data. For one thing, they can be up to 2 kbytes in length (as opposed to the 1.5-kbyte max of gigabit Ethernet). For another thing, they don't have to carry the packet-acknowledgment overhead required in Ethernet LANs running such protocols as TCP.

Switched Storage

But getting up to speed on the SAN itself is just the beginning. The first big decision for net managers concerns topology—and whether to go with switches or hubs.

Right now, switching is where the action is, as indicated by the growing number of vendors. Ancor Communications Inc. (Minnetonka, Minn.), Brocade Communications Systems Inc. (San Jose, Calif.), Gadzoox Networks Inc. (San Jose), McData Corp. (Broomfield, Colo.), and Vixel Switch Operations (formerly Arxcel Technologies Inc., Irvine, Calif.) are the five players now plying the market (see Table 1). But they might soon be joined by Compaq Computer Corp. (Houston), Storagetek Network Systems Group (Brooklyn Park, Minn.), and Sun Microsystems Inc. (Mountain View, Calif.), all of which say they'll release SAN switches within the next year. What's more, most vendors of fibre channel storage hubs are planning to get into switching as well.

But the presence of more vendors doesn't make the decision any easier. There are a lot of issues to consider, starting with the types of network traffic the switch can handle. Like ATM, fibre channel offers specific classes of service. Class 1 creates a direct, acknowledged, connection-oriented fibre channel link. Net managers running apps that require continuous connections, like video clips with synchronized sound, should look for switches with Class 1 capabilities.

Class 2 and Class 3 both cover connectionless links and are the right choice for packetized traffic. The difference between them is that Class 2 also furnishes an acknowledgment that information has been received. Class 3 is similar, but it doesn't issue the acknowledgment. Then again, the value of acknowledging receipt depends on the traffic. "With traffic that's fairly reliable, and with packetized traffic that already contains acknowledgments, you don't buy that much with Class 2 service," says Wayne Rickard, a vice president and general manager with Gadzoox. When should Class 2 be used? When handling online transactions. But for tasks like linking a disk drive and tape backup unit, Class 3 may be just fine.

Two switches—Ancor's Gigworks MKII and Vixel's Rapport 4000—also offer intermix, in which Class 1 and connectionless Class 2 and 3 traffic work together. Here's how intermix could be used: If a set of large files is

Figur

A SAN is video cli switches,

Table 1: Selected Vendors of SAN Switches

Vendor	Product	SAN architectures	Fibre channel topologies	Backplane architecture	Fibre channel classes of service	Fibre channel rates	Media
Ancor Communications Inc. Minnetonka, Minn., 612-932-4000 http://www.ancor.com	Gigworks MKII **Circle No. 471**	Fibre channel	Point-to-point, FC-AL, fabric	Switching matrix	1, 2, 3, intermix	1 Gbit/s	Copper, multim fiber, mixed me (all via GBICs)
Brocade Communications Systems Inc. San Jose, Calif., 408-487-8000 http://www.brocadecomm.com	Silkworm **Circle No. 472**	Fibre channel	Point-to-point, FC-AL, fabric	Switching matrix	2, 3	1 Gbit/s	Copper, single-multimode fiber mixed media (al GBICs)
Gadzoox Networks Inc. San Jose, Calif., 408-360-4950 http://www.gadzoox.com	Denali **Circle No. 473**	Fibre channel	Point-to-point, FC-AL, fabric	Multiplexing bus	2, 3	1 Gbit/s	Copper, single-multimode fiber (both via GBIC
McData Corp. Broomfield, Colo., 303-460-9200 http://www.mcdata.com	ES-4000 Fibre Channel Enterprise Switch **Circle No. 474**	Fibre channel	Point-to-point, fabric	Multiplexing bus	2, 3	1 Gbit/s	Multimode fiber
Vixel Switch Operations (formerly Arxcel) Irvine, Calif., 714-753-9480 http://www.arxcel.com	DS-1000 **Circle No. 475**	Fibre channel	Point-to-point, fabric	Did not disclose	2, 3	266 Mbit/s	Multimode fiber
	Rapport 4000 **Circle No. 476**	Fibre channel	Point-to-point, FC-AL, fabric	Did not disclose	1, 2, 3, intermix	1 Gbit/s	Copper, single-multimode fiber mixed media (all GBICs)

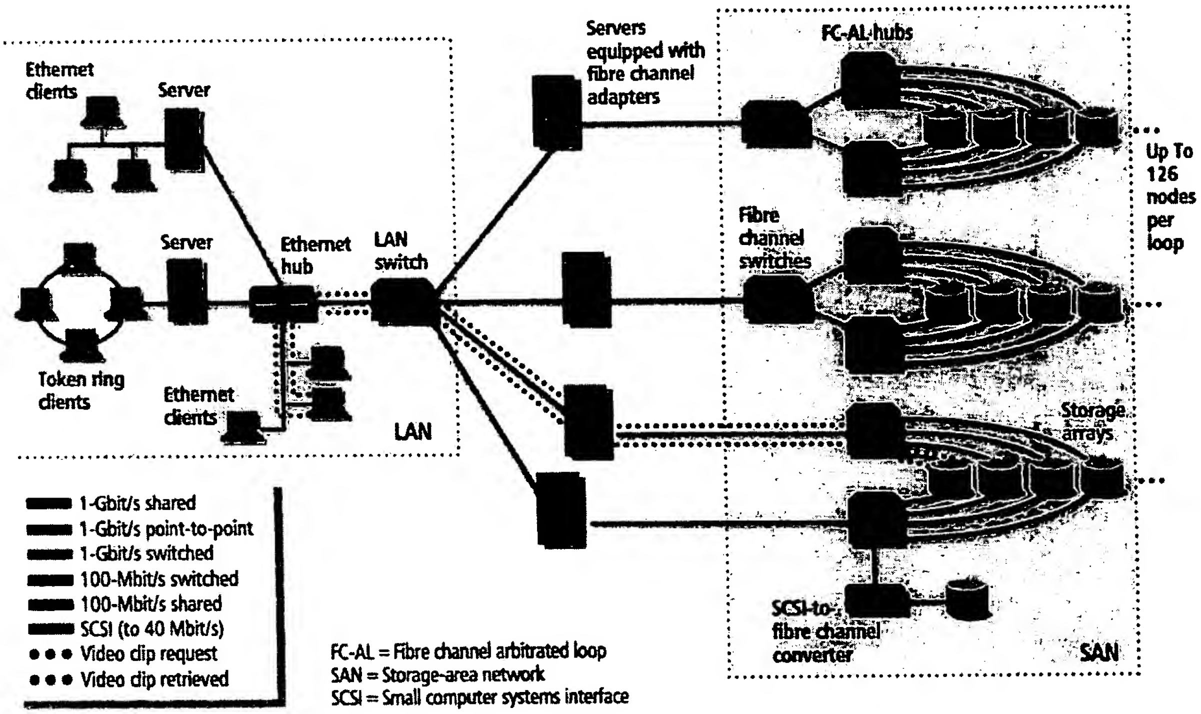
FC-AL = Fibre channel arbitrated loop *GBIC = Gigabit interface converter*

annel link.
hat require
video clips
10uld look
ibilities.
cover con-
ight choice
ference be-
o furnishes
mation has
iilar, but it
nent. Then
edging re-
With traffic
packetized
knowledg-
much with
Rickard, a
manager
Class 2 be
ie transac-
ing a disk
lass 3 may

Gigworks
1000—also
lass 1 and
l 3 traffic
intermix
rge files is

Figure 1: Anatomy of a SAN

A SAN is a dedicated, parallel network that permits high-speed access to multiple storage arrays, where such items as video clips might be archived. It's typically constructed around 1-Gbit/s fibre channel connections and has its own switches, hubs, and gateways, which are linked to the LAN via servers and hosts on the intranet.



Fibre channel rates	Media	Port density	Cascading	Supports third-party storage arrays	Directory included	Multicasting	Management	Price
1 Gbit/s	Copper, multimode fiber, mixed media (all via GBICs)	16 ports per switch	Up to 256 switches	Yes	Yes (via simple name server and proprietary method)	Yes	SNMP agents; Web	$1,995 per port
1 Gbit/s	Copper, single- and multimode fiber, mixed media (all via GBICs)	Up to 16 ports per switch	Up to 32 switches	Yes	Yes (via simple name server)	Yes	SNMP agents; Web	$1,875 per port (GBICs cost extra $130 to $250)
1 Gbit/s	Copper, single- and multimode fiber (both via GBICs)	3 ports per switch	Up to 256 switches	Yes	Yes (via simple name server)	Yes	Element manager software	$12,500 for base unit
1 Gbit/s	Multimode fiber	32 ports per switch	Up to 3 switches	Yes	Yes (via simple name server)	Yes	SNMP console, agent; Web	$4,625 per port (includes management software, console, and GBICs for multimode fiber; extra GBICs $130 to $250)
266 Mbit/s	Multimode fiber	16 ports per switch	Up to 16 switches	Yes	No	No	Proprietary	$2,625 for 16-port configuration
1 Gbit/s	Copper, single- and multimode fiber, mixed media (all via GBICs)	Up to 8 ports per switch	Unlimited	Yes	No (available April 1998)	Yes	Element manager software; Web	$2,300 per port

being transferred, a Class 1 link can handle the transfer itself while a Class 2 or 3 message alerts the host to set up the next transfer, reducing the time of the total transaction.

Another factor to weigh is how the switch shunts data over its backplane, since this can affect capacity. Ancor and Brocade, for instance, say their boxes use switching matrix backplanes. Net managers can get a fix on the capacity available with such an architecture by multiplying the number of connections on the switch's internal matrix by the speed that's offered. Based on these calculations, Ancor's Gigworks MKII and Brocade's Silkworm each offer 16 Gbit/s capacity.

In contrast, switches from Vixel, Gadzoox, and McData feature backplanes in which traffic is arbitrated on and off the switch using time-division multiplexing. To calculate the capacity of these switches, users need to multiply the number of bits the bus is capable of handling at once by the speed at which it operates, in MHz. So the Vixel and Gadzoox switches, which each feature a 32-bit bus running at 33 MHz, offer overall capacity of just over 1 Gbit/s. McData did not disclose the size and speed of its bus.

Net managers interested in switches may also want to look into such added features as multicasting. In other words, can traffic that's coming into the switch be copied and exported to multiple outgoing ports simultaneously? The obvious advantage of this is that it saves time (and trouble) when undertaking massive file transfers or batch-mode backups, which would otherwise have to be done individually.

Every switch but the DS-1000 from McData offers multicasting—but don't think for a minute that multicasting translates into Layer 3 capabilities. "Fibre channel is the highway, the physical medium, we use to transmit any number of protocols," says Jack Rubinson, director of product management at Brocade. "Routing would require snooping packets, and when you start that, you affect speed and defeat the whole purpose of having a storage network."

What net managers may be more interested in are vendor efforts to bring switch prices under control. Right now, for instance, Vixel's 8-port Rapport 4000 costs $
3-port Den
McData's 3
port. The 1
and Brocad
per port.

And pe
the story. P
most likely
GBICs (gig
small transc
channel devi
copper, mul
mode fiber.
tors of swit
more solid
ternal medi
conspicuous
They also
to $400 per

Table 2: Selected Vendors of SAN Hubs

Vendor	Product	Technology	Interfaces	Data rates	Media	Ports	Cascading	Node bypass	Works with third-party arrays	M
Atto Technology Inc. Amherst, N.Y., 716-691-1999 http://www.attotech.com	Accelnet FC **Circle No. 477**	Fibre channel	FC-AL, point-to-point	1-Gbit/s full duplex	Copper; multimode fiber via MIA	5	Yes, up to 126 hubs	Yes; manual and automatic	Yes	P W ap
Gadzoox Networks Inc. San Jose, Calif., 408-360-4950 http://www.gadzoox.com	Gibraltar CM and GS **Circle No. 478**	Fibre channel	FC-AL, point-to-point	1-Gbit/s full duplex	CM: multimode fiber; GS: copper or multimode fiber via GBIC	CM: 10; GS: 12	Yes, up to 13 hubs	Yes; manual and automatic	Yes	S W V
	Bitstrip **Circle No. 479**	Fibre channel	FC-AL, point-to-point	1-Gbit/s full duplex	Copper; multimode fiber via MIA	9	Yes, up to 18 hubs	Yes; manual and automatic	Yes	N
Gigalabs Inc. Sunnyvale, Calif., 408-481-3030 http://www.gigalabs.com	Jigsaw 8 **Circle No. 480**	Switched SCSI (all types)	SCSI, ATM, HIPPI	640 Mbit/s per slot (up to eight slots)	Single- and multimode fiber	8 slots, each with up to 16 ports	Yes, up to 15 hubs	N/A	Yes	W G vi
G2 Networks Inc. Los Gatos, Calif., 408-399-3800 http://www.g2networks.com	2x5 Hub **Circle No. 481**	Fibre channel	FC-AL, point-to-point	1-Gbit/s full duplex	Multimode fiber	10	Yes, up to 5 hubs	Yes; manual and automatic	Yes	Ja
Hewlett-Packard Co. Palo Alto, Calif., 415-857-1501 http://www.hp.com	Shortwave and Longwave FC-AL Hub **Circle No. 482**	Fibre channel	FC-AL	1-Gbit/s full duplex	Single- and multimode fiber via GBIC	10	Yes, up to 2 hubs	Yes; manual and automatic	No	S
Sun Microsystems Inc. Mountain View, Calif., 415-960-1300 http://www.sun.com	Array A5000 FC-AL Hub **Circle No. 483**	Fibre channel	FC-AL	1-Gbit/s full duplex	Single- and multimode fiber via GBIC	7	No	No	No	N
Vixel Corp. Bothell, Wash., 206-806-5509 http://www.vixel.com	Rapport 1000 **Circle No. 484**	Fibre channel	FC-AL, point-to-point	1-Gbit/s full duplex	Copper, single- and multimode fiber; mixed via GBIC	7	Yes, unlimited	Yes; automatic	Yes	N
	Rapport 2000 **Circle No. 485**	Fibre channel	FC-AL, point-to-point	1-Gbit/s full duplex	Copper, single- and multimode fiber; mixed via GBIC	12	Yes, unlimited	Yes; manual and automatic	Yes	W ba ap in H

HIPPI = High-performance parallel interface MIA = Media interface adapter N/A = Not applicable SCSI = Small computer systems interface

PHOTOGRAP

ch-mode
e have to

DS-1000
asting—
at multi-
capabili-
ıway, the
transmit
ays Jack
ıct man-
g would
ıd when
ıeed and
having a

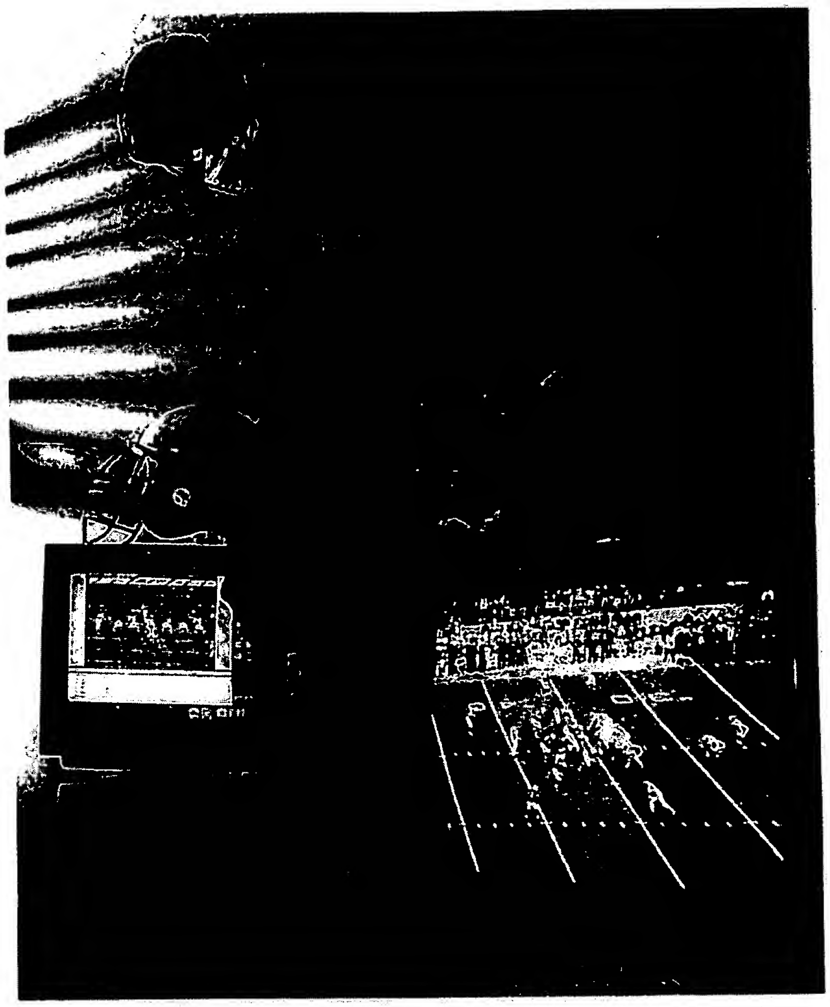
be more
; to bring
ıght now,
Rapport

4000 costs $2,300 per port, Gadzoox's 3-port Denali $4,200 per port, and McData's 32-port ES-4000 $4,625 per port. The 16-port boxes from Ancor and Brocade cost $1,995 and $1,875 per port.

And per-port prices are only part of the story. Prospective buyers also will most likely need to factor in the cost of GBICs (gigabit interface converters)—small transceivers that plug into a fibre channel device in order to adapt ports to copper, multimode fiber, and single-mode fiber. GBICs fit into the form factors of switches and so are considered more solid and reliable than those external media converters that dangle conspicuously from network devices. They also cost more—typically $150 to $400 per port. Some vendors figure the GBIC price into the per-port cost of the switch (like Ancor), while others charge extra (Brocade).

There's something else to keep in mind about switches. Right now, there's concern that the fibre channel spec is weak on how switches should track network devices. It requires the setup of a database (called a simple name server) listing the IP addresses of each storage device and host attached to the switch. Devices in a SAN perform a routine login against the simple name server to join the network. But most vendors would like to see more detail added to the simple name server, such as location information and the protocols supported by various devices. The Fibre Channel Association, which helps define standards for international approval, is looking into improvements, but no specific goals or deadlines have been set.

ıg	Node bypass	Works with third-party arrays	Management	Price
•	Yes; manual and automatic	Yes	Proprietary Windows NT application	$1.195
•	Yes; manual and automatic	Yes	SNMP and Windows NT Ventana	CM: $9,950; GS: $5,800 for base unit, $60 for copper GBIC, $475 for shortwave optical GBIC
›	Yes; manual and automatic	Yes	None	$2,695
›	N/A	Yes	Windows NT Gigaview, or via browser	$2,000 to $8,000 per port
›	Yes; manual and automatic	Yes	Java SNMP	$4,995
›	Yes; manual and automatic	No	SNMP	Shortwave: $9,500; Longwave: $29,500
	No	No	None	$2,300
l	Yes; automatic	Yes	None	$2,650; $500 for short-wave optical GBIC, $1,500 for longwave
l	Yes; manual and automatic	Yes	Windows NT-based SNMP application integrated with HP Openview	$8,000; $500 for short-wave optical GBIC, $1,500 for longwave; $1,250 for management module



Catching the highlights Avid's David Grandin relies on SAN technology to furnish his professional sports customers with simultaneous access to videos stored on multiple disk arrays.

A Handle on Hubs

The arrival of more vendors into the SAN switch market will probably help bring prices down, but customers who can't put their SAN plans off until then may want to consider hubs (see Table 2).

Then again, going with hubs means paying a different price. Hubs share available bandwidth. Although most conform to FC-AL topology, which means they can theoretically handle up to 126 nodes, many vendors say response time sags noticeably when just five or six arrays are attached. Of course, a lot

PHOTOGRAPH BY RICK FRIEDMAN/BLACK STAR

depends on the applications being run. Video clips, for instance, require more bandwidth than data files.

In such cases, corporate networkers should identify those spots where an additional hub might be added (cascaded). For example, if there are two or three huge arrays associated with a particular server or cluster of servers, then a second hub might be necessary between the servers and the storage devices. But if it gets to the point where the number of hubs is affecting performance (and by many accounts that number is pretty low—no more than several), then it may be time to think of adding a switch.

There are other hub issues to keep in mind. Like token ring, fibre channel requires all nodes in a logical loop to respond to switch polls. If a node doesn't respond, the entire loop malfunctions. But most hub vendors offer an automatic bypass feature for failed nodes, which means that a nonresponding node won't bring the entire loop down.

Several hub vendors—Gadzoox and Vixel Corp. (Bothell, Wash.) among them—offer a manual bypass feature. This is helpful in pulling a device out of the loop for reconfiguration or maintenance. The node can then be returned to the SAN, all without stopping the rest of the network. All it requires is making a selection on the configuration interface menu.

Still, when it comes to hubs, price is one of the most attractive features. Gadzoox and Vixel offer 12-port hubs that cost roughly $1,000 per port, depending on media and management options. Gadzoox and G2 Networks Inc. (Los Gatos, Calif.) each offer 10-port hubs, for about $1,000 or less per port. At the low end, Sun and Vixel offer unmanaged 7-port hubs for about $350 per port.

But as with switches, GBICs will have to be factored into the overall hub price. The 12-port Gadzoox Gibraltar GS costs roughly $550 per port if users buy $60 copper GBICs for each port. But the price can climb to $950 per port if multimode fiber GBICs are added; they cost $475 apiece.

What's the first big SAN decision? Whether to go with switches or hubs.

One more thing about SAN hubs: Not every vendor is convinced that fibre channel is the way to go. Gigalabs Inc. (Sunnyvale, Calif.) runs SCSI over ATM to furnish a fast link between remote sites. "There's no need for users to switch to fibre channel," says Kon Leong, president of Gigalabs. "We've made a killer application of simple backup." Still, the Gigalabs hub delivers speeds of just 640 Mbit/s (as opposed to 1 Gbit/s), and it has to be shared among SAN devices. Still, Leong is clear that

resistance t
doesn't mea
out of hand
interfaces w
Jigsaw swit
users will h
SCSI and fi

Wheth
chosen, net
zations may
channel test
ing to mark
help in maki
Among the
ment are A
Calif.), Fini
Calif.), and
(Hampshire

Of co
work also n
dors of SA
varying deg
cade, for inst
sible SNMP
and G2 does

Other

Table 3: Selected Vendors of SAN Internetworking Gear

Vendor	Product	Description	Protocols supported over fibre channel	Interfaces	Ports	Managemen
Advanced Digital Information Corp. (ADIC) Redmond, Wash., 425-881-8004 http://www.adic.com	FCR 100 and 400 Fibre Channel Routers **Circle No. 486**	Packetizes SCSI for transmission over fibre channel network	SCSI	Fibre channel, SCSI	FCR 100:1 fibre channel and 1 SCSI port; FCR 400: 2 fibre channel and 4 SCSI ports	Integral SNMP
Atto Technology Inc. Amherst, N.Y., 716-691-1999 http://www.attotech.com	Fibrebridge **Circle No. 487**	Stackable converter (fibre channel to UltraSCSI)	SCSI	Fibre channel, SCSI	1 fibre channel port; 2 UltraSCSI buses	Windows NT application sup Universal Seria proprietary app
Computer Network Technology Corp. (CNT) Maple Grove, Minn., 612-797-6000 http://www.cnt.com	Ultranet Storage Director **Circle No. 488**	Gateway links hubs, switches, and storage devices over LANs and WANs	SCSI, IP	ATM OC3, Escon, FC-AL, SCSI, T3/E3	Up to 11 slots can handle up to 44 SCSI, 11 FC-AL, or 8 Escon ports	Windows-based application or ir SNMP agent
Crossroads Systems Inc. Austin, Texas, 512-349-0300 http://www.crossroads.com	Crosspoint 4100 and 4400 **Circle No. 489**	Converter (fibre channel to SCSI)	SCSI	Fibre channel, SCSI	4100: 1 fibre channel port, 1 SCSI bus; 4400: up to 2 fibre channel ports and 4 SCSI buses	Integral SNMP
General Signal Networks Inc. Shelton, Conn., 203-926-1801 http://www.gsnetworks.com	IFS/9000 with FX **Circle No. 490**	Converter (fibre channel to SCSI)	SCSI	Fibre channel (single- and multimode fiber only)	4 or 6 fibre channel or SCSI ports	Proprietary pac
Hewlett-Packard Co. Palo Alto, Calif., 415-857-1501 http://www.hp.com	Fibre Channel SCSI Multiplexer **Circle No. 491**	Converter (fibre channel to SCSI)	SCSI	Fibre channel, SCSI	1 fibre channel port, 4 SCSI ports	No
Impactdata Monrovia, Calif., 818-359-4491 http://www.impactdata.com	Network Peripheral Adapter **Circle No. 492**	File server (links SCSI arrays to fibre channel and other networks)	SCSI	Fibre channel, SCSI, HIPPI, ATM, Ethernet, FDDI	2 fibre channel ports, 7 SCSI ports	Proprietary

Gibraltar
rt if users
each port.
0 per port
re added;

AN hubs:
nced that
. Gigalabs

SCSI over
: between
d for users
says Kon
s. "We've
of simple
ub delivers
opposed to
ed among
clear that

resistance to fibre channel migration doesn't mean he rejects the technology out of hand: He says that fibre channel interfaces will be added to the Gigalabs Jigsaw switch this year, at which point users will have the choice of using both SCSI and fibre channel.

Whether switches or hubs are chosen, net managers in large organizations may want to invest in the fibre channel test equipment that's now coming to market. Such gear can be a big help in making configuration decisions. Among the vendors selling test equipment are Ancot Corp. (Menlo Park, Calif.), Finisar Corp. (Mountain View, Calif.), and Xyratex International Ltd. (Hampshire, U.K.).

Of course, building another network also means managing it, and vendors of SAN switches and hubs offer varying degrees of help. Ancor and Brocade, for instance, furnish browser-accessible SNMP agents with their switches, and G2 does the same with its hubs.

Other vendors, however, are less generous. Vixel, for example, charges $1,250 extra for a hardware module that works with the vendor's element management app or with HP Openview. On the plus side, Vixel says one management package is sufficient for as many as 16 hubs.

online extras

Whether looking for serious help or just light on the lingo, the Web is the place to turn for more **storage-area network** details.

- Visit the site of the National Committee for Information Technology Standards (NCITS—pronounced "Insights" by those in the know) to get the scoop on international efforts to reach consensus on a range of technological issues, including storage, SCSI interconnection, and multimedia networking. *http://www.dpt.com/t11/*
- The Fibre Channel Association's site contains information on the consortium's efforts to promote the use of fibre channel technology in members' products. It also features simple, clear explanations of the technology and standards. *http://www.fibrechannel.com*
- The FCLC (Fibre Channel Loop Community) includes nearly 100 vendors of fibre channel storage products. Go to this site to see what they say about SANs. *http://www.fcloop.org*
- The NSIC (National Storage Industry Consortium) includes some 60 U.S. manufacturers of storage products. Visit this site to see what they're doing to promote the cause of SAN research. *http://www.nsic.org*
- Need an overview of storage standards efforts, research projects, historical information, white papers, and trade association data? It's here at Quantum Corp.'s site. *http://www.quantum.com*
- Not convinced that SCSI is through? This site gives up-to-date information on efforts to advance and promote the small computer systems interface. *http://www.scsita.org*
- Looking for links to the IEEE, ANSI, and other groups involved in defining specs used in SANs? Here's the hotline. *http://www.siemens-fo.com/T11.htm*
- This site, run by SNIA (Storage Network Industry Association), features information on the consortium's efforts to standardize various aspects of storage networking. *http://www.snia.org*
- Some industry sources say the VIA (Virtual Interface Architecture) spec backed by power trio Compaq, Intel, and Microsoft will be the key to managing and securing data in storage networks of the future. Judge for yourself by visiting this site. *http://www.viarch.org*

	Management	Price
channel ; FCR 400: and 4 SCSI ports	Integral SNMP agent	FCR 100: $5,300; FCR 400: $30,500
port; 2 s	Windows NT application supports Universal Serial Bus proprietary app	$3,000
in handle l FC-AL,	Windows-based application or integral SNMP agent	$80,000 to $250,000 with 2 fibre channel modules, 2 SCSI modules, and 1 Escon module; management app costs $3,500
nnel port, 0; up to ports and	Integral SNMP agent	4100: $5,995; 4400: starts at $25,000 for 1 fibre channel port and 2 SCSI buses
nel or	Proprietary package	FX module: $9,000; IFX/9000 cabinet starts at $12,000
port,	No	Base unit: $10,400; SCSI card: $1,295; fibre channel card: $3,800
ports,	Proprietary	$40,000 for fibre channel or SCSI configuration

There and Back

Another SAN subject net managers should look into is the conversion between SCSI and fibre channel devices. Fortunately, several vendors now offer SAN internetworking gear, including Advanced Digital Information Corp. (ADIC, Redmond, Wash.), Atto Technology Inc. (Amherst, N.Y.), and Crossroads Systems Inc. (Austin, Texas) (see Table 3). And most products aren't limited to one or two ports, the way most fibre channel adapters are. Rather, they can link groups of SCSI devices (typically seven per bus) to a fibre channel switch or hub. They work by adding a header or other notation to a SCSI transmission in order to shunt it over the fibre channel network.

How does SAN internetworking gear rate when it comes to cost? The answer depends on the size of the network. High-end adapters that furnish a port-to-port link between one host and one fibre channel storage array are typically priced at $1,995. But if four adapters are needed to link a server to four arrays, the cost is $7,980—and four server slots are taken up. In contrast, Crossroads' Crosspoint 4100, which comes with one fibre channel link and one SCSI bus, can connect as many as seven devices to a server—and it costs only $5,995.

Most internetworking gear is intended for use on LANs only, but that's changing too. The Ultranet Storage Director from Computer Network Technology Corp. (CNT, Maple Grove, Minn.), for instance, can extend fibre channel SANs over fast WAN links, including ATM OC3 (155-Mbit/s) circuits and T3 (45-Mbit/s) leased lines (see "Widening the Warehouse," January 1998; http://www.data.com/hot_products/ultranet.html).

Right now it's the only product that offers WAN connectivity, but things could change by late this year, when ANSI is expected to approve a WAN tunneling technique for use in fibre channel networks.

The Rest of the Gear

So if net managers have by now decided that SANs are in the plans, they might be wondering whether their current storage devices can work with the fibre channel switches and hubs being introduced. When it comes to disk arrays, at least, the news is good: Most products comply with the fibre channel standard. Leading vendors Fujitsu Ltd. (Tokyo), Quantum Corp. (Milpitas, Calif.), and Seagate Technology Inc. (Scotts Valley, Calif.) offer links to fibre channel controllers on their drives, which means networkers can simply plug in their switches and hubs and get going.

Makers of tape libraries and optical storage jukeboxes—like ADIC and Tandberg Data Inc. (Simi Valley, Calif.)—are reportedly readying fibre channel tape drives, but at press time none had shipped.

Net managers also can assemble their own fibre channel arrays using stacks of disks from different vendors along with third-party controllers. These "just a bunch of disks" (JBOD) configurations are easy to assemble and relatively cheap. An alternative to JBOD arrays are so-called intelligent or all-in-one arrays. These tend to offer better performance because they're optimized for use on fibre channel networks. Among the all-in-one vendors are Megadrive Systems (Chatsworth, Calif.) and Raidtec Corp. (Alpharetta, Ga.).

Specialized servers that are built with SANs in mind also are emerging. These devices combine arrays with software and hardware designed just for storage applications. Artecon Inc. (Carlsbad, Calif.), for instance, packs multiple processors, RAID arrays, an NFS server, and operating system software into a single box. The advantage? According to the vendor, it's cheaper than dealing with multiple arrays and servers, and it offers a boost in performance, as well.

And a handful of vendors sell host adapters that can furnish fibre channel connectivity for all types of servers. These include Adaptec Inc. (Milpitas, Calif.), Emulex Corp. (Costa Mesa, Calif.), Interphase Corp. (Dallas), Jaycor Networks Inc. (San Diego), Strategies/Qlogic Corp. (Costa Mesa, Calif.), and Sun. Prices typically range from $595 to more than $3,000.

Finally, some vendors are addressing efficiency by equipping their standalone servers with file system software. Tricord Systems Inc. (Plymouth, Minn.) is one of them. "We try to reduce the number of backup requests going over the network in order to accommodate SAN configurations," says Joan Wrabetz, vice president and chief technical officer.

The vendor is still testing its software but says it can increase performance on SANs, where file I/O isn't slowed by the handling of multiple packets. ■

REQUEST FOR COMMENT
If you'd like Data Comm *to publish more articles on this subject, please circle 462 on the Reader Service Card.*

Your Frame Relay network is growing at unprecedented rates. Great! But so is the complexity of that network: SVCs, voice over Frame Relay, even the sheer volume of PVCs. Increased complexity means the increased probability of problems. The solution? Frame Relay testing with TREND'S Aurora TEMPO.

TEMPO:
- Tests both PVCs & SVCs.
- Is cheaper, smaller and easier to use than any other similar device. (there is no similar device)
- Ping test. Powerful SVC bulk calls and more.
- IS FUN to hold in your hands.

don't wait for a DeNniS to show up, get a TEMPO

TREND

www.trendcomms.com

Circle 72 on Reader Service Card

AS 5

Press Releases

HEWLETT PACKARD

Search

HP's High-speed I/O Business Unit

August, 1997

Hewlett-Packard Company is committed to delivering quality high-speed serial I/O products and solutions for gigabit-plus systems, focusing on the mass-storage, high-speed-networking and high-speed-interconnect markets. Today, HP is a leader in Fibre Channel solutions; tomorrow, HP expects to be a leader in other high-speed I/O solutions.

Situation Analysis

The proliferation of data-farming and workgroup clustering is driving demand for reliable, high-speed I/O technology. Heightened user demands include instant access to information, increasingly data-intensive PC video, imaging and Internet applications and advanced system architectures. In addition, the need for advanced data-storage technologies is assuming greater urgency as current storage capacity becomes exhausted and existing I/O technology reaches its limitations.

Fibre Channel technology, which overcomes the limitations of existing I/O technology, offers OEMs and users industry-leading performance, connectivity and scalability. It is firmly positioned to become the industry's primary server-storage interface.

Fibre Channel is an advanced data-transfer technology providing performance levels that meet graphics and video requirements while introducing networking ideas to storage attachment. By merging high-speed I/O and networking functionality into a single connectivity solution, Fibre Channel minimizes I/O bottlenecks, enabling the rapid movement of data through the storage network.

HP's Solution

HP is addressing the increasing need for advanced Fibre Channel technology through its High Speed I/O Business Unit within the Communications Semiconductor Solutions Division of HP's Components Group. The High Speed I/O Business Unit, established in 1996, is furthering HP's leadership in open-market high-speed I/O components through the development of a broad family of industry-leading Fibre Channel protocol and physical layer components.

The High Speed I/O Business Unit also is HP's center of development for I/O subsystems, such as 32- and 64-bit PCI host bus adapters, with associated software device drivers.

As a whole, HP's Components Group is the world's largest independent supplier of communications components, with 12,500 employees worldwide. The group, through the development of semiconductor solutions that enable the information-exchange revolution, is advancing such strategic technologies as the global information infrastructure, the extended desktop and mobile information appliances.

The Communications Semiconductor Solutions Division is focused on delivering industry-leading communications technology for emerging markets with high growth potential. These markets include wireless communications, Fibre Channel links for switch and data storage, synchronous optical network (SONET), gigabit Ethernet, asynchronous transfer mode (ATM) and digital cellular phones, cordless phones, pagers and wireless communications for portable computers.

HP and Fibre Channel

For nearly 10 years, HP has been involved in creating, developing and promoting Fibre Channel standards and technology.

In 1988, when the American National Standards Institute chartered the Fibre Channel Working Group to develop a practical, inexpensive, expandable method of transferring large volumes of information very quickly among workstations, mainframes, supercomputers, desktop computers, storage devices and display devices, HP took an active role in developing a workable standard. Since that time, HP has continued to define physical and signaling interface standards for the Fibre Channel market.

Since the creation of the Fibre Channel Systems Initiative (FCSI) and the Fibre Channel Association (FCA) in 1993, HP has been active in these industry organizations, supporting the development and implementation of Fibre Channel technology.

FCSI, a joint effort between HP, IBM and Sun Microsystems, has helped advance Fibre Channel as an affordable, interoperable, high-speed interconnection standard for workstations. Through this initiative, HP has helped develop application-specific models that aid designers in the creation of Fibre Channel products, easing the transition to Fibre-Channel technology.

HP also has participated in the efforts of the FCA to create a support structure for system integrators, peripheral manufacturers, software developers, component manufacturers, communications companies and computer service providers that encourages the implementation and utilization of Fibre Channel technology.

HP also is a founding member of the Fibre Channel Loop Community, which supports designers and developers in implementing Fibre Channel Arbitrated Loop technology for storage-attachment solutions.

Major Product Areas

Today, HP Components Group and the High-Speed I/O Business Unit offer their customers one of the broadest portfolios of Fibre Channel products available, delivering a single-vendor source for high-performance Fibre Channel technology.

In 1992, HP introduced the first in its gigabit-link chip sets, the first silicon IC chip sets designed for high-speed point-to-point communication at rates of up to 1.5Gb/s. That year, HP also introduced a 266Mb/s Fibre Channel optical-link card, fully implementing the FC-0 physical layer of the Fibre

Channel standard.

In 1995, HP released the TACHYON Fibre Channel controller IC to customers for development. Since its release, the TACHYON IC has been designed-in by more than 30 OEMs, capturing the largest share of the emerging Fibre Channel controller market. With its established level of maturity, interoperability and broad market acceptance, the TACHYON IC is the de facto standard for Fibre Channel controllers.

HP's TACHYON Fibre Channel controller IC features a complete hardware-based implementation that is optimized for the high-performance characteristics Fibre Channel offers over other I/O technologies. With the TACHYON architecture, customers have demonstrated sustained throughputs in excess of 1Gb/s and an industry-leading I/O rate of 10,500 I/Os per second.

HP also has developed a line of component-level products to address the Fibre Channel Physical Interface (FC-PH) of the Fibre Channel standard developed by the American National Standards Institute. This line includes high-speed fiber optic modules (FC-0) serializer ICs, fully integrated FC-0 gigabit-link modules (GLMS), and in Q4 gigabit integrated cards (GBICs).

Current Products

- HPFC-5000 TACHYON Protocol Chip single-chip Fibre Channel interface (no I/O processor required); supports 266MBd, 531MBd, and 1,062.5MBd links; supports three topologies: direct connect, fabric and Fibre Channel; arbitrated Loop (FC-AL); and supports Fibre Channel class 1, 2, 3 and Fibre Channel intermix services.

- HDMP-1526 FC-0 integrated transmitter IC 1,062.5MBd data rate; TTL-compatible with 5V power supply; low power, 14 x 14 mm package; and 10-bit-wide interface.

- HDMP-1512 / HDMP-1514 FC-0 Layer Integrated Circuits FC-0 integrated transmitter / receiver IC; selectable for 531.25MBd and 1,062.5MBd data rates; selectable on-chip laser driver or 50-ohm cable driver with transmitter; selectable on-chip equalizer on receiver for improved BER and distance with 50-ohm coaxial cable; TTL-compatible with single 5.0V supply; and 20-bit-wide interface.

- HGLM-1063 FC-0 Daughter Card Products full FC-0 layer gigabaud link modules for 531MBd and 1,062.5MBd data rate; compatible with FC-0 layer; compatible with "gigabaud Link Module Specification;" TTL-compatible with single 5.0V supply; and Class 1 laser safety.

- HFBR-5301 / HFBR-5302 FC-0 PMD Layer Transceivers Fibre Channel multimode fiber duplex SC transceiver for 133MBd and 266MBd links up to 1.5 km; and optoelectronic compliance to Fibre Channel 12-M6-LE-I / 25-M6-LE-I interface.

- HFBR-1119T / HFBR-2119T FC-0 PMD Layer Transmitters/Receivers Fibre Channel multimode fiber ST connectored transmitter /receiver for 266MBd links up to 1.5 km; DLT1102-FC Fibre Channel singlemode fiber FC connectored transmitter; 266MBd links up to 2 km; and optoelectronic compliance to Fibre Channel 25-SM-LL-I interface.

Future Directions

HP believes that, with continued proliferation of data-intensive applications and increasing demand for higher-capacity networks and immediate access to information, the need for high-performance I/O technology will only become greater. Fibre Channel technology, which is designed to grow with the needs of users, offers OEMs a scalable solution. Future generations of Fibre Channel technology are expected to offer even greater I/O performance, meeting users' growing needs for high-speed networking capabilities.

Press Releases
HEWLETT PACKARD
Search
Feedback to the WebMaster

(c) Copyright 1997 Hewlett-Packard Company.

AT5

6/9/3 (Item 3 from file: 810)

DIALOG(R)File 810:Business Wire

(c) 1999 Business Wire . All rts. reserv.

0732312 BW0074

HEWLETT PACKARD: *HP*'S *TACHYON* Fibre Channel Controller Chosen by Compaq; Industry-leading IC to Enable Next-generation Storage Solutions

August 06, 1997

Ticker Symbol:	HWP
Byline:	Business Editors & Computer Writers
Dateline:	PALO ALTO, Calif.
Time:	08:00 PT
Word Count:	655

PALO ALTO, Calif.--(BUSINESS WIRE)--Aug. 6, 1997--Hewlett-Packard Company today announced that Compaq Computer Corporation (Houston) has selected HP's industry-leading *TACHYON* Fibre Channel controller IC for Compaq's next generation of storage solutions.

Fibre Channel is an advanced data transfer technology providing performance to meet graphics and video application requirements while introducing networking ideas to storage attachment. Merging high-speed I/O and networking functionality in a single connectivity solution, Fibre Channel incorporates many of the features found in networking, such as interconnection through hubs and switches. This combination minimizes I/O bottlenecks, enabling the rapid movement of data through the storage network.

HP's *TACHYON* controller IC features a complete hardware-based implementation optimized to take advantage of the performance advantages of Fibre Channel technology. In customer implementations, the *TACHYON* IC architecture has delivered sustained throughputs of 1 Gb/s and an industry-leading I/O rate in excess of 10,500 I/Os per second.

"Compaq is committed to delivering the highest-performing products to our customers," said Alan Skidmore, director of engineering for Compaq's Enterprise Storage and Options Division. "The expanded power and functionality offered by Fibre Channel and by HP's *TACHYON* Controller IC enable us to provide storage solutions with the stability, high-bandwidth operation and increased rates of data transfer necessary for today's demanding computing environment."

"HP's *TACHYON* solution allows OEMs to implement Fibre channel solutions easily," said Thomas Lahive, senior analyst with Dataquest. "The allocation of resources from HP and Compaq will drive broad industry acceptance of this new technology. With a common backbone shared between network and storage architectures, users will have the ability to access information at rates never imagined before."

"Since its release in 1995, the *TACHYON* IC has become the overwhelming choice of OEMs seeking the highest level of Fibre Channel performance and design stability," said Julian Elliott, business unit manager of HP's High-speed I/O Business Unit. "Compaq's endorsement reinforces the leadership of the *TACHYON* IC and is a significant milestone in establishing Fibre Channel as a broadly adopted standard. We look forward to our continued partnership with Compaq in enabling this emerging market."

HP first released the *TACHYON* IC to customers for development in early 1995. Since its release, the *TACHYON* IC has been incorporated by more than 30 OEMs, capturing the largest share of the emerging Fibre Channel-controller market. Because of its level of maturity, interoperability and broad market acceptance, the *TACHYON* IC now is the de facto standard for Fibre Channel controllers.

The *TACHYON* IC supports a broad range of Fibre Channel features:

-- three data rates -- 1062.5, 531, and 266MBd links;
-- three topologies -- Arbitrated Loop (FC-AL), Fabric and Point-to-point;
-- all classes of service -- Class 1, 2, 3 and Intermix; and
-- supports existing I/O protocols, including SCSI and IP.

HP AND FIBRE CHANNEL

HP has been very active in the creation, development and promotion of Fibre Channel standards and technology for nearly 10 years. Working within numerous Fibre Channel industry organizations, including the Fibre Channel Working Group chartered by the American National Standards Institute, the Fibre Channel Systems Initiative (FCSI), the Fibre Channel Association (FCA) and the Fibre Channel Loop Community, HP has played an active role in supporting the development and implementation of Fibre Channel technology and industry standardization.

HP offers a broad range of Fibre Channel products, from components- to system-level solutions.

Information in this release applies specifically to products available in the United States. Product availability and specifications may vary in non-U.S. markets.

If you choose to review these items, your readers will receive the quickest response to their inquiries by mailing them to Hewlett-Packard Company, Components Response Center, 3175 Bowers Ave., M/S 88U Santa Clara, Calif. 95054-9929.

CONTACT: Hewlett-Packard Company
Northe Osbrink, 408/435-6765
northe osbrink@hp.com
or
Copithorne & Bellows for HP
Laura Ruthenbeck, 415/975-2233
laura.ruthenbeck@cbpr.com

KEYWORD: CALIFORNIA
INDUSTRY KEYWORD: COMED COMPUTERS/ELECTRONICS TELECOMMUNICATIONS INTERACTIVE/MULTIMEDIA/INTERNET PRODUCT

AR6

ProQuest®

« Back to Document View

Databases selected: ProQuest Direct Complete

What's new

Business Wire
The Global Leader in News Distribution

HP's TACHYON Family of Fibre Channel Protocol Controllers to be Part of Compaq's Next-generation Storage Solutions

High Tech Writers. **Business Wire**. New York: Dec 7, 1998. pg. 1

Author(s):	High Tech Writers
Dateline:	*CALIFORNIA*
Publication title:	Business Wire. New York: Dec 7, 1998. pg. 1
Source type:	Wire feed
ProQuest document ID:	36556504
Text Word Count	501
Document URL:	http://proquest.umi.com/pqdweb?did=36556504&sid=2&Fmt=3&clientId=5409&RQT=309&VName=PQD

Abstract (Document Summary)

"Compaq's strategic direction for high-performance and high-capacity storage will leverage Fibre Channel technology," said Bob Schultz, vice president, Server Storage Business Unit at Compaq. "The Compaq and HP TACHYON Fibre Channel architecture forms an important part of our Fibre Channel storage solutions. Compaq will utilize the 64-bit TACHYON in several storage systems and host-bus adapters, providing Fibre Channel storage solutions for the work group through the enterprise data center."

"Compaq's selection of TACHYON PCI-Fibre Channel protocol controllers is an important extension of our relationship, enabling the Fibre Channel market to build momentum," said Julian Elliott, manager of HP's High-Speed I/O Products Business Unit. "HP and Compaq share a common vision of Fibre Channel as an important new storage I/O technology for creating TACHYON-based products that are the most interoperable, highest performing Fibre Channel solutions. The TACHYON TL is the first such controller." TACHYON Benefits

Full Text (501 words)

Copyright Business Wire Dec 7, 1998

PALO ALTO, Calif.--(BUSINESS WIRE)--Dec. 7, 1998--Hewlett-Packard Company today announced that Houston-based Compaq Computer Corporation has selected HP's latest 64-bit PCI version TACHYON protocol controller IC for a future storage product.

"Compaq's strategic direction for high-performance and high-capacity storage will leverage Fibre Channel technology," said Bob Schultz, vice president, Server Storage Business Unit at Compaq. "The Compaq and HP TACHYON Fibre Channel architecture forms an important part of our Fibre Channel storage solutions. Compaq will utilize the 64-bit TACHYON in several storage systems and host-bus adapters, providing Fibre Channel storage solutions for the work group through the enterprise data center."

Compaq's adoption of the controller IC is an important step in the growth of Fibre Channel high-speed storage. The company is the market-share leader in open, multiuser storage solutions, according to the September 1998 International Data Corporation Worldwide Disk Systems Market Forecast and Review.

"Compaq's selection of TACHYON PCI-Fibre Channel protocol controllers is an important extension of our relationship, enabling the Fibre Channel market to build momentum," said Julian Elliott, manager of HP's High-Speed I/O Products Business Unit. "HP and Compaq share a common vision of Fibre Channel as an important new storage I/O technology for creating TACHYON-based products that are the most interoperable, highest performing Fibre Channel solutions. The TACHYON TL is the first such controller." TACHYON Benefits

The TACHYON TL protocol controller is optimized for mass-storage applications, including host-bus adapter cards and storage subsystems. It is FC-AL-2 compliant, supports Class 3 and Class 2 ACK-0 services, is capable of simultaneous initiator and target functionality, and operates in both private and public loops. Its ability to directly attach to a fabric in N-port mode sets it apart from competing controllers.

The TACHYON TL protocol controller delivers leading-edge performance. In recent testing, it achieved more than 31,000 I/Os per second, the highest I/O rate achieved to date by any Fibre Channel protocol controller. Its high-performance features include fully automated SCSI command processing, full duplex support with simultaneous inbound and outbound frame processing, and multiple inbound sequence support. This combination delivers up to three times the performance of competing controllers in switched environments.

The TACHYON TL protocol controller is highly interoperable, leveraging the mature, de facto standard TACHYON architecture used by more than 50 OEM suppliers of Fibre Channel solutions and Compaq's PCI architecture (used in Compaq computers).

Because the TACHYON TL protocol controller does not require companion SRAM, it can be used to implement solutions that are more cost-effective because they use fewer components and occupy less board area. About HP

Hewlett-Packard Company is a leading global provider of computing, Internet and intranet solutions, services, communications products and measurement solutions, all of which are recognized for excellence in quality and support. HP has 124,600 employees and had revenue of $47.1 billion in its 1998 fiscal year.

Information about HP and its products can be found on the World Wide Web at http://www.hp.com/.

Information about HP's Fibre Channel components and solutions can be found on the World Wide Web at http://www.hp.com/go/fibrechannel. SEQN: BW0185

Copyright © 2005 ProQuest Information and Learning Company. All rights reserved. Terms and Conditions

Text-only interface

ProQuest COMPANY

factiva

Dow Jones & Reuters

BYTE

COVER STORY: TECH OUTLOOK '98

Serving Up Storage Faster SCSI and Fibre Channel SANs set the stage for servers that run and run.

By Scott **Mace**
505 words
1 January 1998
BYTE
72
Vol. 23, No. 1
English

(Copyright 1998 McGraw-Hill, Inc.)

Disk I/O subsystems, not CPUs, are the bottlenecks in today's servers. However, that will change in 1998.

The changes start with the venerable SCSI, which is suddenly doubling in speed and length, going from a 40-MBps burst rate to 80 MBps with PCI Ultra2 SCSI. At the same time, Adaptec has released technology to increase the length of SCSI cables from 3 meters to 12 meters.

The speed improvement will help servers keep up with processor improvement, while preserving investments in previous SCSI hardware. Ultra2's 25-meter distance will help drive disk storage out of the confines of the server cabinet itself and into rack-mounted external RAID and more exotic subsystems. Mixing and matching external disk to server will become commonplace. And next year, Adaptec will be sampling Ultra3 technology, doubling SCSI's burst rate again to 160 MBps.

But SCSI is just the beginning. Fibre Channel, an ANSI-standard network that can multiplex both SCSI and IP traffic, extends 30 meters over copper wires or as far as 10 kilometers on fiber-optic cables. It is capable of speeds in excess of 100 MBps in both directions. In 1998, expect to see host adapter offerings from HP, Compaq, and others. Fibre Channel networks will come to resemble Gigabit Ethernet networks, employing hubs and switches as the core of server farms.

Overlaying both SCSI and Fibre Channel are emerging Storage Area Networks (SANs), such as those from Computer Network Technology and Tricord. SANs not only separate storage nodes from server nodes on networks, they also let different servers share a common pool of data. SANs let users expand disk capacity without having to bring down application servers.

Mere speed and capacity are one thing, but affordable disaster recovery and scalability are the holy grails of the data center. Unix systems have offered clustering technology, with automatic failover to backup servers, for years. Disk mirroring has been a feature of Novell's SFT III for almost as long. This year, Microsoft's Server Cluster option brings to Windows NT 4.0 similar reliability features. After NT 5.0 ships, Microsoft will expand Server Cluster to support more than two nodes, and NT will gain the scalability of those nodes working together to share the load of applications, such as database servers, which are written to take advantage of Cluster Server.

Not to be outdone, the next release of Novell's NetWare, known as Moab, will be able to support 16-server clusters when its Orion option, formerly known as Wolf Mountain, ships. Novell is promising Orion for the second half of 1998.

Illustration: High-speed storage subnetworks can employ SCSI, Fibre Channel, and wide-area media such as ATM or DS3 connections. Illustration: Adaptec Milpitas, CA 408-945-8600 http://www.adaptec.com Computer Network Technology Minneapolis, MN 800-268-0090 http://www.cnt.com

Document byte000020020327du110080g

© 2005 Dow Jones Reuters Business Interactive LLC (trading as Factiva). All rights reserved.

AT 6

[_private/smsheader.htm]

November '97 Article

Fibre Channel Speeds the Convergence of Network and Storage

by Carla Kennedy

Today the compute and data communications architecture of most companies resembles a patchwork of old and new technologies, patched together in a frantic attempt to keep up with faster computers, increased traffic and growing storage demands. Yet for all of the brainpower, man-hours and money that get thrown at them, many network and storage managers struggle to keep up with, let alone get ahead of, growing data demands and changing user requirements.

While IT managers search for solutions (and while users drum their fingers waiting for large data files to move from storage to server to workstation), new data communications technologies are laying a new path out of the chaos: one that begins with core elements of existing networking and storage management and evolves to elegant new architectures that promise incremental performance improvements, simpler configuration and implementation, and lower cost of ownership.

In many ways, the future is now: the building blocks of this infrastructure embrace existing protocols and standards-based technologies now available.

The current state of the enterprise

Current network topologies (and storage management departments) typically are defined by political walls separating "storage" and "network" functions. Whether SCSI vs. IP or channel vs. network, this structure may have served vendors well, but also have introduced too many roadblocks, bottlenecks and potential blackouts to effectively meet changing customer needs – let alone effectively optimize the compute horsepower available in today's typical workstations and servers.

The problems this polarized structure can cause are multiple: for starters, common network and storage management architectures simply haven't kept pace with rapid increases in computing power and dramatic increases in the volumes of data stored and carried in a typical networked environment. Users are demanding more out of their computers and the applications they run on them. Storage requirements and demands are increasing at an exponential rate - doubling every six months in some businesses. And, network performance is being throttled by the winding client-to-server-to-storage-and-back path these large data files must follow in a typical I/O subsystem to reach the desktop and compute servers.

The moral of the story is that current configurations simply won't get users where they need to be. The question is, what will?

The roadmap indicates a move to a heterogeneous distributed compute architecture incorporating centralized, network-attached storage and compute clusters and linked with a high-performance Fibre Channel fabric. This architecture offers a number of key advantages including improved performance, connectivity, scalability, more efficient utilization of compute resources and ease of implementation. With Fibre Channel as the fabric, the old barriers between storage and networking can be replaced by a new seamless compute infrastructure that provides a high-speed interconnect, whether carrying IP, SCSI or any other high-level protocol.

Fibre Channel: a storage/server/workstation fabric interconnect

A new generation of Fibre Channel networks has been deployed that illustrates the power and promise of this new infrastructure. In Japan, the world's largest Fibre Channel network deployed earlier this year by one of the world's leading consumer and industrial goods manufacturers demonstrates the bandwidth and scalability of Fibre Channel.

This network, currently comprised of 800 IBM and Silicon Graphics workstations and servers, is linked using a fabric of 40 Ancor GigWorks™ Fibre Channel switches and designed to support data-heavy communications associated with CATIA, a computer-aided design and modeling application.

Data file sizes, peak load requirements and the sheer size of the installation drove the decision to deploy Fibre Channel at this site. A single CATIA model file size can range from 5 to 10 MB, and assembly model files are 80 to 100 MB in size. The traffic spikes at the beginning of the day, over lunch and at the end of the day when engineers move files to or from servers and workstations would easily overwhelm other options like ATM and Fast Ethernet, but are no problem for the network, which claims an aggregate bandwidth of more than 300 gigabytes for the first phase alone. Also, the size of the installation – when complete, this site will include 1,800 workstations and servers linked with an Ancor Fibre Channel fabric – was far beyond the scaling capabilities of Gigabit Ethernet or ATM. Although IP, a standard networking protocol, is used to transport the files, the problem being solved has the same characteristics of a typical server-to-storage bottleneck problem.

But ultimately, for this customer there was a compelling business case for installing Fibre Channel: CATIA lets this company's engineers do much of their prototyping and testing right on the screen, rather than building expensive models. This short cut can shave months or even years off of product development cycles, giving the company a significant "time-to-market" advantage over the competition. Eliminating costly hand-built prototypes saves money, too.

Fibre Channel server-storage interconnect

Fibre Channel already is building market momentum as a storage interconnect. Ancor's GigWorks switches and adapters are providing access to storage in the radiology department at the University of California, Los Angeles medical school, supporting the transfer of x-ray images (which typically run 30 megabytes in size) between Sun UltraSPARC servers and mass storage devices. With transfer times reduced from 90 seconds to about 8 seconds, this application also suggests the potential of telemedicine, where high-speed interconnects will let specialists at remote locations review patient files and images. This promises better patient care, lower costs and streamlined operations for care providers. According to Lu Huang, senior technical manager for UCLA Radiology Imaging and Information Systems, using Fibre Channel as the connection between storage devices and the server "is like having your storage on a local bus."

At first glance, these customer sites don't appear to have much in common. One is a large IBM/Silicon Graphics environment with files moving between servers and workstations. The other is a relatively small (fewer than 10 workstations) Sun environment using Fibre Channel for file transfer between server and storage. But when one compares a simplified network configuration of each environment, except for the protocol they're running they're virtually identical. One is a traditional local area network, the other is a SAN, or "system area network." In each, the Fibre Channel fabric provides the high-speed interconnect between servers and nodes.

Where the enterprise is headed, and how to get there

Integrators such as EDS recognize the importance of forward thinking in developing data communications architectures that can help the company and its clients maximize their investment in compute infrastructures and people. Just as vendor business interests drove the evolution of current network architectures, client business interests will drive the move to a new seamless compute infrastructure. Trends in storage and system interconnects suggest that high performance Fibre Channel fabrics can be a key component of this infrastructure.

The foundation of a more heterogeneous data communications model can be seen in EDS' PIPE (Primary Integrated Platform Environment) program. PIPE represents both a strategic architectural vision and a business initiative designed to help EDS maintain its leadership in the integration and computer services industry. The PIPE program includes products from a variety of vendors - all thoroughly tested and certified compliant with the PIPE architectural vision. Ultimately, PIPE is a branded suite of products which EDS consultants will recommend for customer sites.

New generation fabrics yield streamlined architecture

This model shows how a protocol-independent interconnect like Fibre Channel can blur the lines between the traditional storage and client "sides" of the server. For example, switched or arbitrated loop Fibre Channel interconnects offered on new-generation products like the GigWorks MKII switch offer optimal scalability, letting users add storage or compute resources as their requirements dictate, and often without adding costly additional servers or storage subsystems. This configuration provides simplified installation and management as well.

The products now coming to market are designed to support two data communications concepts that are building blocks of flattened data communications architectures: compute clustering and network-attached storage.

Rapidly increasing network traffic and the fast-growing number of data-intensive applications now available to users are driving the shift to a new distributed computing model. With terabytes and even petabytes of data being stored and moved across many networks today, the storage community is leading the way toward a new revolutionary system/storage architecture, with key elements borrowed from networking. This architecture will provide greatly improved access to data from all components of a distributed compute environment.

With network management costs averaging approximately 40 percent of the typical annual network budget, there are significant resource and management savings to be realized from this model as well: Centralized storage resources are easier to house and manage, and require just a fraction of the server processing power of de-centralized storage configurations.

A Fibre Channel overview

Fibre Channel was created to blend the simplicity, speed and reliability of channels with the flexibility of networks – making it well suited as a high-performance fabric supporting both LAN and storage communication.

This gigabit technology is protocol-independent, mapping channel and network traffic into Fibre Channel frames. This lets Fibre Channel handle SCSI operations between the fabric and RAID storage devices, or a file transfer from one TCP/IP device to another simultaneously over the same fabric. And with data bundled in variable-size frames up to 2112 bytes, Fibre Channel provides an efficient, flexible and low-overhead structure for optimal throughput. In addition, Fibre Channel sequencing allows TCP/IP to send 64K bytes with a single CPU I/O for more efficient compute utilization.

By definition, networks need to be flexible to accommodate the needs of different users. Applications, traffic and purpose vary from user to user across the enterprise, meaning "one size fits all" approaches to data communications simply won't deliver optimal solutions in production environments. Consider ATM, a technology that proponents have tried to position as optimal for video, voice and data. ATM uses fixed 53 byte frame sizes (standard in the telephony industry), but a single uncompressed film frame represents about 40 megabytes of data. Clearly, a connectionless service like ATM presents significant problems in supporting error-free transmission in this type of data-intensive environment – particularly when leaky-bucket flow control schemes are the only safeguard.

On the other hand, Fibre Channel addresses SCSI hardware shortcomings, offering increased connectivity, scalability and performance. SCSI is limited to 16 devices per channel, where Fibre Channel arbitrated loop will support up to 126 devices. SCSI is limited to cable lengths of 25 meters where Fibre Channel can run distances of up to 10 kilometers between nodes. And where SCSI is limited to transfer rates of 40 Mbytes-per- second (with plans to increase to 80 Mbytes-per-second), Fibre Channel currently offers 100 Mbytes per second transfer rates with plans to push that rate to 400 Mbytes-per- second in the future.

Fault tolerance becomes even more critical when network functions are directly tied to the operations of the company. Fibre Channel offers the flexibility of three distinct classes of service to handle data communications in storage and workgroup LAN applications:

Class 1 is a circuit-switched service that sets up a dedicated connection with dedicated bandwidth to guarantee data delivery – ideal for transmitting large visualization or video data files in a client/server environment, for example.

Class 2 and 3 are connectionless services, better suited to smaller messages and general data traffic, where latency becomes an issue that could impact overall performance. Running SCSI over Class 2/3 Fibre Channel in a storage/server environment, for example, delivers optimal small message performance and data integrity to support mission-critical applications like investment account management for a bank or brokerage firm, or a company's payroll.

Intermix combines Class 1, 2 and 3 messages across the fabric to maximize network capacity. All services can be multiplexed over the same circuit, but only Class 1 connections get guaranteed bandwidth. Class 2 and 3 frames are slipped in during "idles."

Two-dimensional switching delivers optimal performance in Fibre Channel Intermix. The Ancor GigWorks switch uses a circuit switch for Class 1 traffic, and a separate packet switch for Class 2 and 3 traffic. Fibre Channel frame headers carry the routing instructions. For transmission of large files using Class 1 service, only the header on the first frame is read. Once the end point is determined and the circuit established, the entire message is transmitted. This scheme results in extremely low latencies – approximately 500 nanoseconds for Ancor's newest-generation switch.

Moving forward: the ripple effect

In today's market, compute infrastructures are more than just a physical asset. They're critical to the operations, productivity and profitability of the business. And today's technology decisions will ripple through the enterprise for years to come. In the end, these decisions will either position the enterprise for a smooth transition to even better future performance, or for a turbulent ride through the choppy waters of technologies that failed to deliver as promised. New heterogeneous compute architectures using ANSI- standard Fibre Channel for the data communications fabric are designed to help IT managers navigate these waters with maximum confidence.

Carla Kennedy is the Vice President of Marketing at Ancor Communications, Inc. (Minnetonka, MN). http://web.archive.org/web/19991117133021/http://www.ancor.com/.

[_private/smsfooter.htm]

AR7

RAB™

Computer Technology Review March 1998

The Most Definitive Coverage of Disk Array/RAID Technologies

RAID In Clusters

By Vincent Fleming

The practice of clustering systems has increased in popularity over the past few years, and appears to be gaining momentum. This is not surprising, considering that clustering offers an alternative to very large computer systems that is both cheaper and more reliable.

Reliability is one of the major benefits of building clusters. If all the systems in a cluster need to be online for the cluster to operate, the Mean Time Between Failures (MTBF) would be unbearably short and reliability poor. This is analogous to using RAID 0 in a data storage array. The MTBF of a RAID 0 group with 10 drives is 10 times less than that of a single drive—because if one drive fails, the entire group goes offline. The same would be true for clusters without redundancy.

Cluster Redundancy

The many different clustering applications solve the reliability problem in the same way—software that allows one of the other Nodes in the cluster to take the place of a failed Node. This is only possible if both Nodes can access each other's disk drives. As an example, consider a 2-Node cluster running two databases—one on each Node. If Node A goes down, Node B needs to be able to access Node A's database—which just happens to be on Node A's disks. So, how can one Node access another (dead) Node's disks?

The answer is a disk subsystem that can be attached to more than one Node at a time—a feature of cluster-ready RAID subsystems. This is often referred to as Dual Attachment and has the added benefits of high performance and reliability of the disk subsystem.

The general class of RAID subsystems that support Dual Attachment is external, rather than internal, subsystems. They are attached to the host computer (or cluster) via SCSI, fiber channel, or ServerNet. You must ensure that the RAID subsystem used in a cluster supports Dual Attachment, since many do not.

Dual Initiation

The most common method of Dual Attaching a RAID Subsystem is via SCSI. There are two ways that this can be done using standard SCSI. The first is Dual Initiated. A Dual Initiated RAID subsystem is one that is attached to two or more host computers via SCSI, and all hosts are on the same SCSI bus. This is accomplished by assigning each host computer a different SCSI Initiator ID (a disk drive is a target of I/O, so it gets a Target ID: a host computer initiates I/O, so it gets an Initiator ID—which are basically the same thing.) In Fig 1, we could set Node A to have ID 7, Node B to have ID 6, and the disk subsystem to have ID 0 to have a valid configuration. Note that all devices (hosts/disks/arrays) are on the same SCSI bus.

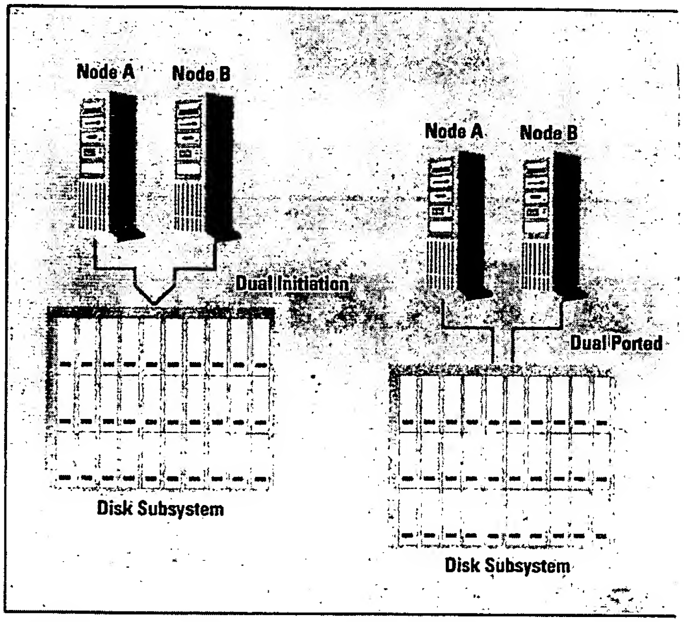
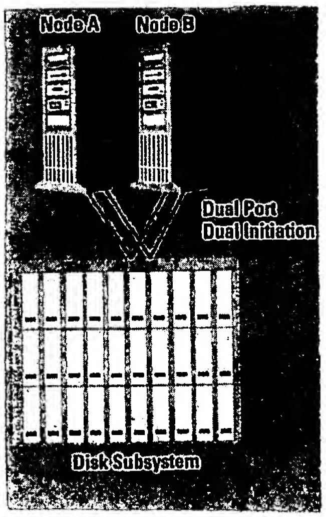


Fig 1 There are two methods of Dual Attaching SCSI devices: Dual Initiating and Dual Porting.

Dual Porting

The second method of Dual Attaching SCSI devices is by Dual Porting. A Dual Ported RAID subsystem is also one that is attached to two host computers via SCSI, but each host has its own SCSI bus. We could set both Node A and Node B to ID 7, and the disk subsystem to ID 0 to have a valid configuration.

While RAID subsystems that support Dual Porting are less common, there are several operational and functional advantages of Dual Porting over Dual Initiating. Dual Porting is by far more robust and efficient than Dual Initiating. Dual Porting is more robust because of its tolerance to bus failures. If the cable between Node A and the RAID subsystem fails, Node B can still access the RAID subsystem. This is not the case with Dual Initiating—should a cable fail, termination of the bus would be lost, and the SCSI bus would no longer operate. Dual Porting also provides added bandwidth: each Node has full SCSI bus bandwidth to the RAID subsystem. With Dual Initiating both Nodes must share the SCSI bus, halving the bandwidth available to each Node!



Continued from page 50

RAID

Dual Port/Dual Init

Combining Dual Porting and Dual Initiating provides further fault tolerance to the cluster by providing two, redundant SCSI buses from each Node to the RAID subsystem. Each Node would use one of the SCSI buses as a Primary path to the RAID subsystem, and the remaining SCSI bus as a Secondary path. Configuring the Nodes so they do not share Primary paths (Node A's Primary path is Node B's Secondary path) allows each Node to have full bandwidth to the RAID subsystem. Adding Host-resident software to manage the failover from Primary to Secondary paths increases Node uptime, and therefore overall cluster reliability. Unfortunately, very few CPU manufacturers support this feature directly and RAID vendor supplied software and drivers are necessary.

Fibre Channel

Fibre Channel is an interconnect technology that can support Dual Attachment of RAID subsystems. By using dual redundant Fibre Channel Arbi-trated Loop (FC-AL) loops, RAID subsystems can attach to multiple hosts, much like the above description of Dual Initiator SCSI configurations. FC-AL has eliminated the single point of failure of Dual Initiated SCSI by adding second loop, which remains active if the primary

See RAID page 52

fails or is disconnected. However, there is nothing analogous to Dual Porting when using Fibre Channel.

ServerNet

ServerNet is yet another interconnect that supports Dual Attachment of RAID subsystems. ServerNet is different from SCSI and Fibre Channel in that it is a switched network fabric, not just a bus. ServerNet incorporates routers, much like TCP/IP routers. Each device has its own redundant connection to the router, and the router passes data and commands from Node to Node, Node to disk, or disk to disk. Because of this routed configuration, ServerNet provides fault tolerance and connectivity that is superior to either Fibre Channel or SCSI (Fig 2).

Finally, there is one other application for RAID in clustered environments that should be mentioned. Since the boot drive of a computer is the most likely component to cause downtime, using a RAID device for this drive can keep the cluster running more smoothly. For example, in Windows NT systems using a PCI-based non-redundant RAID controller can significantly increase the reliability, and even speed, of each individual Node. Other operating systems can take advantage of proprietary and non-proprietary RAID systems, including software RAID (via host-resident virtual device drivers), and SCSI attachable internal RAID devices.

Summary

The use of RAID arrays to protect data from disk failures is well known. In a clustered environment the correct choice of RAID array architecture can add reliability and robustness to the clustered environment. ■

Vincent Fleming *is the director of advanced solutions at ECCS, Inc. (Tinton Falls, NJ).*

www.eccs.com

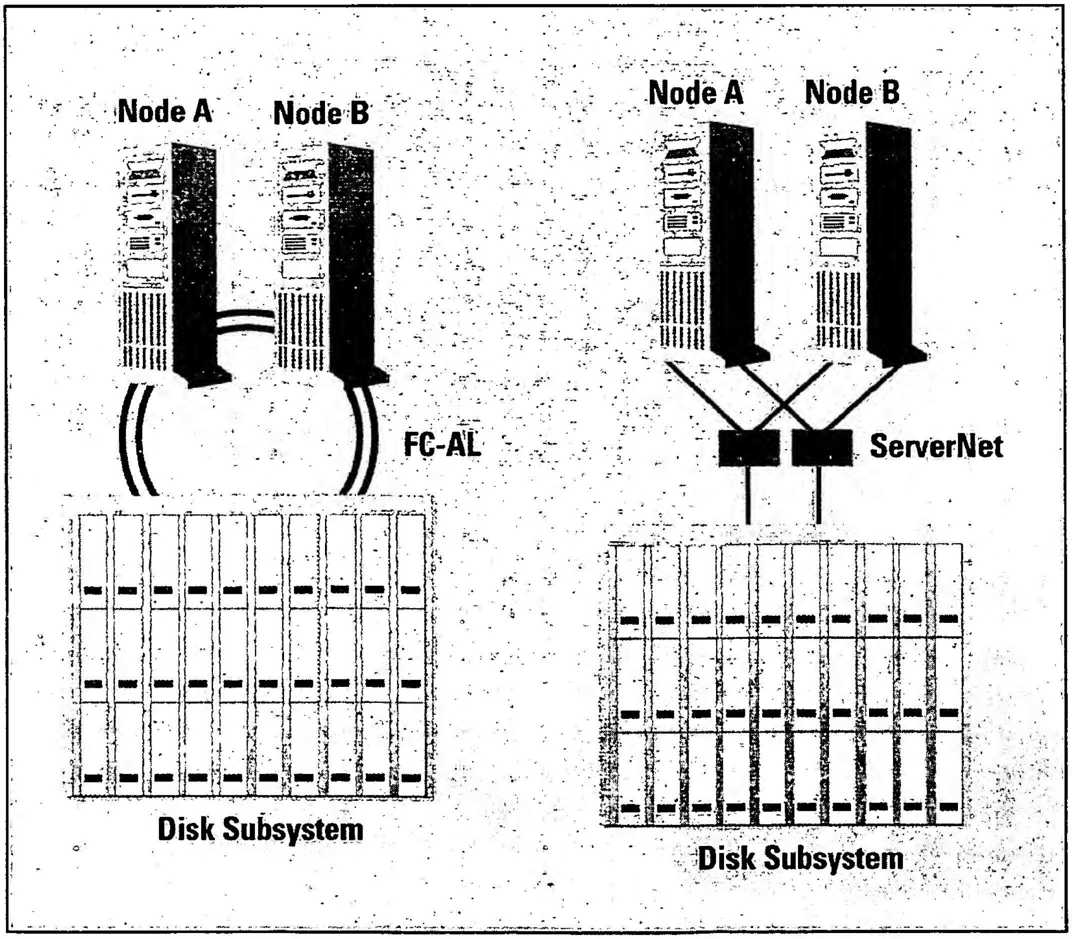


Fig 2 Dual redundant FC-AL loops and ServerNet are two interconnect technologies that can support Dual Attachment of RAID subsystems.

RAID in clusters
Vincent Fleming
Computer Technology Review; Mar 1998; 18, 3; ProQuest Direct Complete
pg. 50

RAID In Clusters

By Vincent Fleming

The practice of clustering systems has increased in popularity over the past few years, and appears to be gaining momentum. This is not surprising, considering that clustering offers an alternative to very large computer systems that is both cheaper and more reliable.

Reliability is one of the major benefits of building clusters. If all the systems in a cluster need to be online for the cluster to operate, the Mean Time Between Failures (MTBF) would be unbearably short and reliability poor. This is analogous to using RAID 0 in a data storage array. The MTBF of a RAID 0 group with 10 drives is 10 times less than that of a single drive—because if one drive fails, the entire group goes offline. The same would be true for clusters without redundancy.

Cluster Redundancy

The many different clustering applications solve the reliability problem in the same way—software that allows one of the other Nodes in the cluster to take the place of a failed Node. This is only possible if both Nodes can access each other's disk drives. As an example, consider a 2-Node cluster running two databases—one on each Node. If Node A goes down, Node B needs to be able to access Node A's database—which just happens to be on Node A's disks. So, how can one Node access another (dead) Node's disks?

The answer is a disk subsystem that can be attached to more than one Node at a time—a feature of cluster-ready RAID subsystems. This is often referred to as Dual Attachment and has the added benefits of high performance and reliability of the disk subsystem.

The general class of RAID subsystems that support Dual Attachment is external, rather than internal, subsystems. They are attached to the host computer (or cluster) via SCSI, fiber channel, or ServerNet. You must ensure that the RAID subsystem used in a cluster supports Dual Attachment, since many do not.

Dual Initiation

The most common method of Dual Attaching a RAID Subsystem is via SCSI. There are two ways that this can be done using standard SCSI. The first is Dual Initiated. A Dual Initiated RAID subsystem is one that is attached to two or more host computers via SCSI, and all hosts are on the same SCSI bus. This is accomplished by assigning each host computer a different SCSI Initiator ID (a disk drive is a target of I/O, so it gets a Target ID; a host computer initiates I/O, so it gets an Initiator ID—which are basically the same thing.) In Fig 1, we could set Node A to have ID 7, Node B to have ID 6, and the disk subsystem to have ID 0 to have a valid configuration. Note that all devices (hosts/disks/arrays) are on the same SCSI bus.

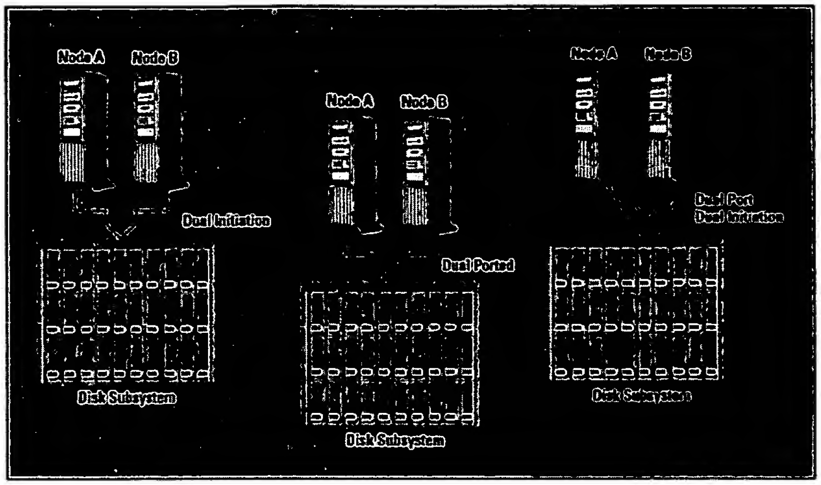


Fig 1 There are two methods of Dual Attaching SCSI devices: Dual Initiating and Dual Porting.

Dual Porting

The second method of Dual Attaching SCSI devices is by Dual Porting. A Dual Ported RAID subsystem is also one that is attached to two host computers via SCSI, but each host has its own SCSI bus. We could set both Node A and Node B to ID 7, and the disk subsystem to ID 0 to have a valid configuration.

While RAID subsystems that support Dual Porting are less common, there are several operational and functional advantages of Dual Porting over Dual Initiating. Dual Porting is by far more robust and efficient than Dual Initiating. Dual Porting is more robust because of its tolerance to bus failures. If the cable between Node A and the RAID subsystem fails, Node B can still access the RAID subsystem. This is not the case with Dual Initiating—should a cable fail, termination of the bus would be lost, and the SCSI bus would no longer operate. Dual Porting also provides added bandwidth; each Node has full SCSI bus bandwidth to the RAID subsystem. With Dual Initiating both Nodes must share the SCSI bus, halving the bandwidth available to each Node!

Dual Port/Dual Init

Combining Dual Porting and Dual Initiating provides further fault tolerance to the cluster by providing two, redundant SCSI buses from each Node to the RAID subsystem. Each Node would use one of the SCSI buses as a Primary path to the RAID subsystem, and the remaining SCSI bus as a Secondary path. Configuring the Nodes so they do not share Primary paths (Node A's Primary path is Node B's Secondary path) allows each Node to have full bandwidth to the RAID subsystem. Adding Host-resident software to manage the failover from Primary to Secondary paths increases Node uptime, and therefore overall cluster reliability. Unfortunately, very few CPU manufacturers support this feature directly and RAID vendor supplied software and drivers are necessary.

Fibre Channel

Fibre Channel is an interconnect technology that can support Dual Attachment of RAID subsystems.' By using dual redundant Fibre Channel Arbi-trated Loop (FC-AL) loops, RAID subsystems can attach to multiple hosts, much like the above description of Dual Initiator SCSI configurations. FC-AL has eliminated the single point of failure of Dual Initiated SCSI by adding second loop, which remains active if the primary

See RAID page 52

■ SCSI 3, Fast/20, Wide
■ Same Feature Set as Ultra2000
■ Prices start under $5,000
■ Use with any host:
■ Laptop or PC
■ Work Station
■ Dumb Terminal

Analyzers

SCSI-3, Fast/20, Wide ■
Self contained, Portable ■
7.5 nsec resolution ■
Initiator Emulation ■
SE, DIFF, & LVD standard ■
Hard Disk Storage ■
Analysis Software ■

Laptop Configuration [illegible] Configuration

Ancot Wrote The Books on SCSI & Fibre Channel Call for FREE "Basics of SCSI", or "What is Fibre Channel" 2nd Ed. or to order "Fibre Channel-Vol. 1-The Basics"

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

RAID

Continued from page 50

fails or is disconnected. However, there is nothing analogous to Dual Porting when using Fibre Channel.

ServerNet

ServerNet is yet another interconnect that supports Dual Attachment of RAID subsystems. ServerNet is different from SCSI and Fibre Channel in that it is a switched network fabric, not just a bus. ServerNet incorporates routers, much like TCP/IP routers. Each device has its own redundant connection to the router, and the router passes data and commands from Node to Node, Node to disk, or disk to disk. Because of this routed configuration, ServerNet provides fault tolerance and connectivity that is superior to either Fibre Channel or SCSI (Fig 2).

Finally, there is one other application for RAID in clustered environments that should be mentioned. Since the boot drive of a computer is the most likely component to cause downtime, using a RAID device for this drive can keep the cluster running more smoothly. For example, in Windows NT systems using a PCI-based non-redundant RAID controller can significantly increase the reliability, and even speed, of each individual Node. Other operating systems can take advantage of proprietary and non-proprietary RAID systems, including software RAID (via host-resident virtual device drivers), and SCSI attachable internal RAID devices.

Summary

The use of RAID arrays to protect data from disk failures is well known. In a clustered environment the correct choice of RAID array architecture can add reliability and robustness to the clustered environment. ■

Vincent Fleming *is the director of advanced solutions at ECCS, Inc. (Tinton Falls, NJ).*

www.eccs.com

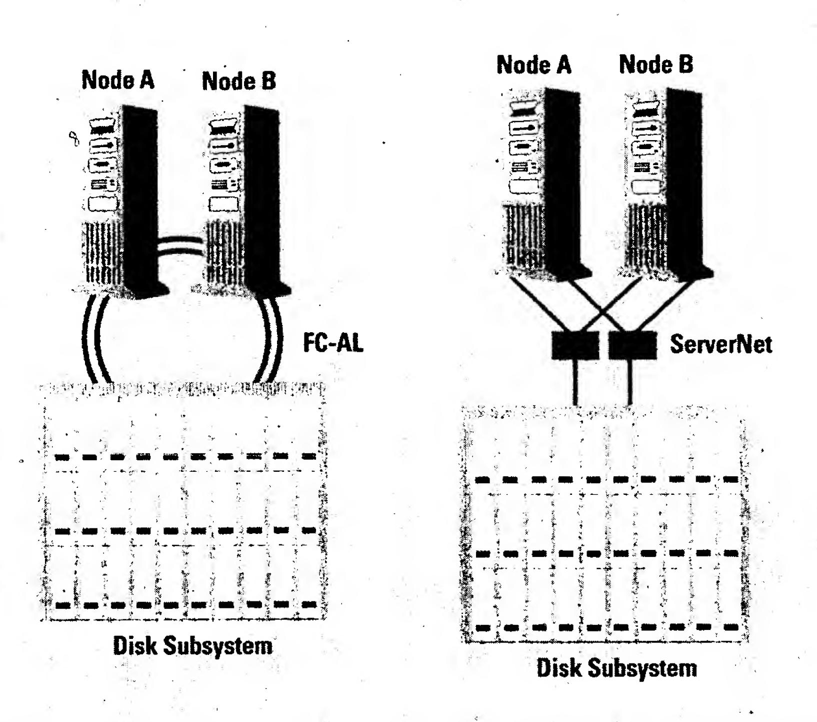
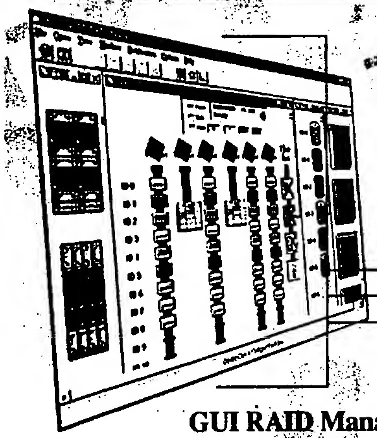


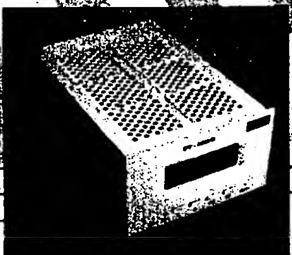
Fig 2 Dual redundant FC-AL loops and ServerNet are two interconnect technologies that can support Dual Attachment of RAID subsystems.

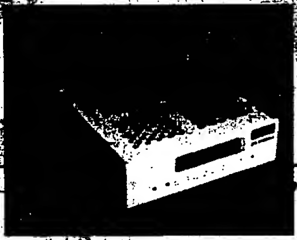
Infortrend

NEW! Hot-Swappable/SCSI-to-SCSI RAID Controllers

Infortrend, a recognized leader in RAID Controller design and manufacturing, introduces three new hot-swappable RAID Controllers.







GUI RAID Manager

- Supports all INFORTREND RAID controllers
- Remote management over networks
- Event notification by e-mail, fax and pager
- Online RAID expansion

IFT-3100U

- Ultra-SCSI to Ultra-SCSI
- Redundant controller capability with synchronous cache
- Expandable to 9 channels
- Fibre and LVD options available Q2/98
- High performance Pentium CPU
- RAID levels 0,1(0+1), 3 and 5

IFT-3102UG

- Ultra-SCSI to Ultra-SCSI
- Expandable to 8 channels
- LVD option available Q2/98
- RAID levels 0,1(0+1), 3 and 5

IFT-3101UG

- Ultra-SCSI to Ultra-SCSI
- Compact 3.5" board
- 210-pin hot-swappable connector design
- 3 channels
- RAID levels 0,1(0+1), 3 and 5

1st Company to Ship...

- 5.25 inch, HH SCSI - SCSI RAID Controllers / February 1994
- 3.5 inch, HH SCSI - SCSI RAID Controllers / June 1995
- Half-length PCI RAID Card / April 1996
- Ultra Wide Solution / August 1996

Infortrend
RAID Controller Innovation

For customers in North, Central and South America:
Infortrend Corporation, 149 Stony Circle, Ste. 210, Santa Rosa, California [illegible], USA
Telephone: (707) 541-3400 Fax: (707) 541-3409 www.infortrend.com sales@infortrend.com

For customers in Asia/Pacific, Europe and UK:
Infortrend Technology Inc., 6F-6, No-351 Chung-Shan Road, [illegible] Sec. Chung-Ho City, Taipei Hsien, Taiwan R.O.C.
Telephone: [illegible]-2226-0126 Fax: 886-2-2226-0020 www.infortrend.com.tw sales@infortrend.com.tw

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

How will migrating to fibre channel occur?
Brian R Smith
Computer Technology Review; Summer 1997; ProQuest Direct Complete
pg. 92

ASI

How Will Migrating To Fibre Channel Occur?

By providing customers a bridging methodology from SCSI and ESCON

by Brian R. Smith
Crossroads Systems, Inc.

Fibre Channel as a technology and the concept of storage networking are on the cusp of becoming widely available. To make the benefits known to the end-user, they must become widely understood. Once understood, the marketplace for storage networking based upon Fibre Channel will see rapid growth and deployment. A transition strategy from the point-to-point nature of storage today to the storage networking paradigm is required to facilitate this transition. How will this occur? It will occur by providing customers a methodology to bridge from their existing storage solutions, SCSI and ESCON in particular, to the future of storage networking based on Fibre Channel.

There have been many articles describing Fibre Channel's capabilities. In summary, Fibre Channel is a hybrid channel and networking data transport medium. It has been an ANSI standard since 1996. Fibre Channel operates at Gigabit speeds, supports storage and networking, and can be switched or interconnected as a shared media. Most of the systems OEMs and peripheral device vendors have made product announcements using Fibre Channel as their migration path for SCSI storage. As a community of infrastructure providers, we now need to describe how storage networking will integrate into customer environments and how those customers will migrate their businesses. A number of vendors are beginning volume shipment of products but a broader view, a systems view, is what is needed to inform and educate an MIS team on the merits and usage of Fibre Channel infrastructure components to solve their day to day business problems.

Today's customer is interested in an improved storage solution, including some of the following requirements: increased distance between server and storage or server and tape backup or both, scaleable storage for all applications and faster back up, and increased, predictable performance in access to user's data. Each of these and additional applications can be supported as an outgrowth of the flexibility and capability of Fibre Channel and storage networking.

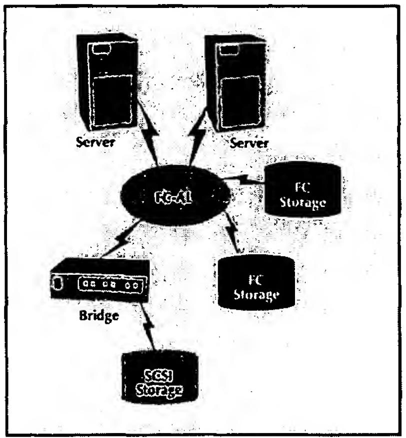


Fig 1 Bridges enable the customer to have SCSI and Fibre Channel storage co-reside on the network.

A key element of migrating from existing storage solutions today, such as SCSI and ESCON, to Fibre Channel is to preserve the investment customers have made in their data and backup infrastructures. To facilitate this, the concept of bridging and

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

routing of storage protocols must be introduced. Bridging and routing are words that are familiar to the internetworking world. A mapping of their definition over to the migration of SCSI and ESCON to Fibre Channel is needed. Bridging is defined as the one to one conversion that takes place between SCSI and Fibre Channel. For instance, converting one SCSI bus to one Fibre Channel N_Port or NL_Port would constitute a bridge. Routing is the conversion between multiple SCSI buses and one or many Fibre Channel N_Ports or NL_Ports. Bridging and routing SCSI to Fibre Channel is the first step in the transition a customer's infrastructure must adopt to reach the realm of storage networking.

At the nuts and bolts level, bridging between SCSI and Fibre Channel is defined as taking the SCSI command (CDB), SCSI data and sense information and converting it over to FCP_CDB, FCP_DATA and FCP_RSP, respectively, on Fibre Channel. There is an associated addressing mapping that must occur as well between SCSI's view of the triplet (bus, target, LUN) and Fibre Channel's FCP_LUN field. The FCP_LUN field is an 8 byte field that can be utilized in several ways. The most common would be to support the SCSI Controller Command (SCC) set view which has long term value. Another method for the short term is to utilize the FCP_LUN field in flat mode where no hierarchical definition is imposed. The key to this discussion for the customer is that solution providers must work together for interoperability at the system level.

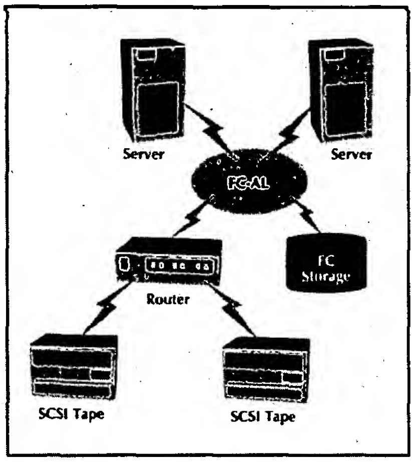


Fig 2 Midrange of the connectivity spectrum would be a router with two SCSI buses on it.

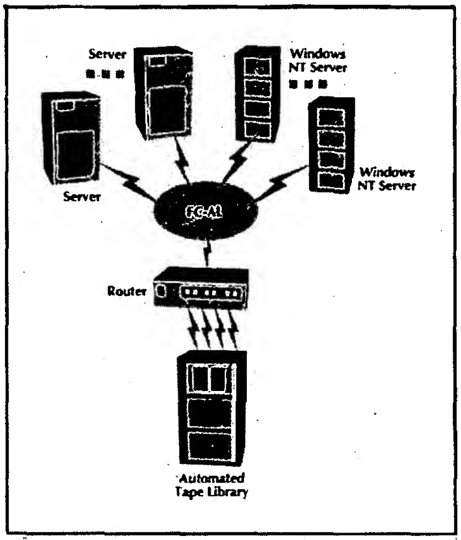


Fig 3 Putting a storage bridge or router on the front-end provides Fibre Channel connectivity at a small percentage of original equipment cost.

This will require many vendors to interoperate, including host bus adapter (HBA) vendors, bridge and router vendors, hub and switch vendors, application vendors, and peripheral vendors, to complete the solution for the customer. The interoperability matrix can be overwhelming for a single infrastructure supplier to support alone. As a solution, our weak link is including the application vendors in our interoperability work. Supporting a systems level integration facility will go a long way to achieving valuable functionality for the customer.

Now that the conversion process has been described, why would a customer implement this conversion? Primarily because they have significant investment in SCSI data and tape backup, and are moving to the domain of Fibre Channel to solve additional storage problems present in their infrastructure today. Four examples of bridging and routing that benefit the customer will be shown in remainder of this paper. These examples are: the migration of disk storage to Fibre Channel from SCSI; access to legacy SCSI devices from new Fibre Channel servers; centralized, automated backup of servers (Win NT, Unix, etc.) over Fibre Channel; and disaster tolerance and recovery with existing SCSI devices. (Fig 1)

As customers add servers and storage to solve their business problems, access to existing data remains vital. The new storage being added is Fibre Channel based, therefore, the issue of communicating with SCSI data will arise. Bridges enable the customer to have SCSI and Fibre Channel storage co-resident on the storage network,

CUT YOUR COSTS TO THE BONE:
Call "The RAID Enclosure Specialist"

• RAID Server Barebones with Mylex or Infortrend RAID Controller and ASUS Motherboard
• 10 to 28 Bay Server Enclosures

1-888-CD-TOWER
(238-6937)
Website: www.constor.com

• RAID Subsystem Barebones with Mylex or Infortrend RAID Controller
• 4 to 28 Bay Subsystem Enclosures

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

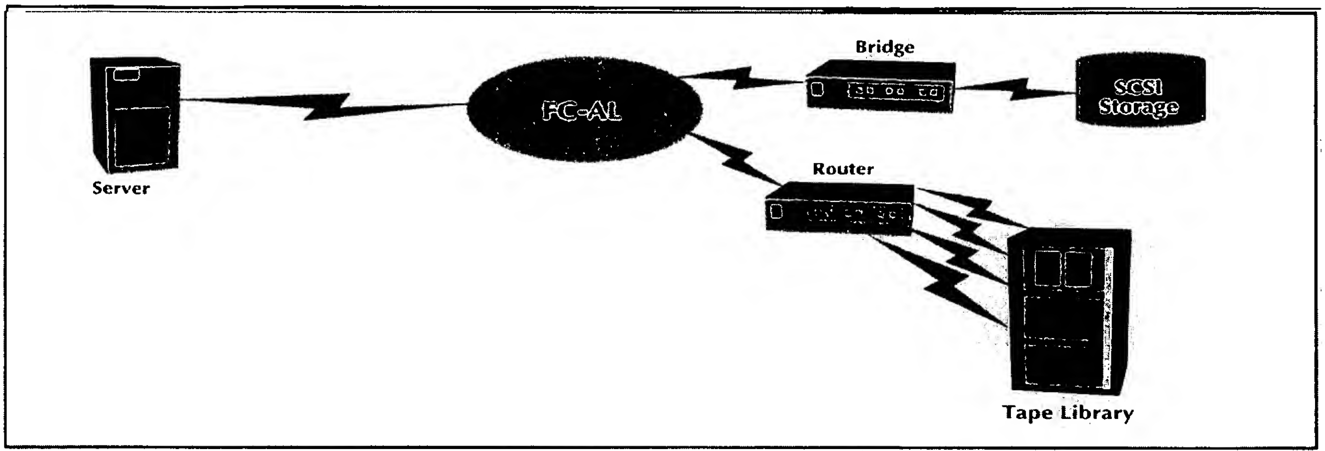


Fig 4 FC-AL's faster backup allows MIS to back-up more data in less time.

thus preserving the data and the investment in that data. With routing, multiple SCSI storage units on multiple SCSI buses can be aggregated onto the storage network (FC-AL or switched) to provide optimal speed matching.

For a customer to choose a bridge, he or she would be adding Fibre Channel connectivity to the server or servers and storage, and moving that new storage onto FC-AL. A bridge then would "front-end" the existing SCSI storage (RAID or JBOD) to bring it onto the storage network. Typically one bridge would be used to interface one SCSI RAID or a tape library to FC-AL. The router provides a similar function to the bridge but for higher end connectivity. An example of a high-end router implementation would be to connect four SCSI RAID units to the storage network through one Fibre Channel port. At the mid-range of the connectivity spectrum would be a router with two SCSI buses on it. This router would typically be used to attach a two or four drive tape library to FC-AL for the purpose of backing up new Fibre Channel storage. (Fig 2)

Next is the ability to satisfy the desire by a customer to utilize existing SCSI devices, particularly the more expensive devices, such as high end tape and optical jukeboxes, in the new paradigm. Customers have purchased expensive SCSI devices over the years and find significant uses for them. These devices will never have a migration path to Fibre Channel. Therefore, putting a storage bridge or router on the front-end to provide Fibre Channel connectivity at a small percentage of the original equipment's purchase price is very attractive. If this device happens to be an old SCSI device, then a bridge or router can keep its slowness from negotiating faster devices down to its level. Overall system speed is therefore improved. (Fig 3)

Third, with the topology of FC-AL, a high-end tape backup system can be built with existing SCSI tape libraries to backup Windows NT, Unix and other servers. By attaching each of the servers to FC-AL, connectivity is made to an automated backup system. In conjunction with an industry standard backup package, this solution becomes very attractive to the MIS manager who is currently backing up tens of servers with tape devices attached individually to each server. The system of an operator switching tapes for each server and starting backups is prone to error. The costs of individual tape devices on each server is significantly higher than using a bridge or router in conjunction with a tape library system. An automated system for server backup is a significant improvement for the entire MIS staff. Beyond the convenience of automated backup is a substantially higher backup bandwidth that storage networking can supply. FC-AL can support over 300 Gigabytes per hour of sustained backup bandwidth, whereas a backup solution using Fast Ethernet would run at only 20 Gigabytes per hour. This faster backup gives the MIS group the ability to back up substantially more data in a shorter amount of time. (Fig 4)

Finally, disaster tolerance and recovery are becoming key requirements for today's enterprise customer. Placing a copy of the backed up data off-site most frequently involves carrying tapes to another location miles away. This process makes data recovery painfully slow and inefficient. If the data could be backed up and restored in real-time from a remote location, the customer would improve productivity and have a disaster tolerant solution. Given that Fibre Channel nodes can communicate over 10 kilometers, duplicate storage devices and backup systems can be placed at significant distances from the servers that access them. By using a bridge or router to "front-end" the remote SCSI device(s), full Fibre Channel bandwidth is available to a remote, secure location. Having a remote dynamic copy of data is becoming critical to business success around the world.

Storage bridging and routing brings additional value that can benefit the MIS customer. One is expansion of their server slots, essentially providing a higher level of connectivity for the server. This is particularly applicable for PCI servers with limited slots. An additional benefit is facilitating more flexible server architectures to solve specific customer problems with general purpose hardware. ■

Brian R. Smith *is Co-Founder and CEO of Crossroads Systems, Inc. (Austin, TX).* **www.crossroads.com**

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

AT7

SCSI applications on Fibre Channel

Robert Snively

Sun Microsystems Computer Corporation
2550 Garcia Ave., Mountain View, CA 94043-1100

ABSTRACT

The Fibre Channel is a general purpose point-to-point serial connection standard permitting a variety of transmission media, including both copper and fiber optic transmission options with bit rates ranging from 266 megabits per second up to 1065 megabits. A switching fabric is defined for the Fibre Channel to allow the interconnection of large numbers of nodes at high bandwidth. The Fibre Channel standard defines the implementation from the physical media and connectors up through switched and connectionless transport level services well suited for high performance data transfers.

For the attachment of disk and tape storage subsystems, a communication model is required to define the command, data transfer, and response sequences that access the subsystem. The Small Computer System Interface (SCSI) provides an internationally standardized architectural model optimized for the attachment of storage subsystems and other intelligent devices to host computers. This paper describes the mapping of SCSI onto Fibre Channel and describes the architectural advantages of SCSI as a model for communication across the Fibre Channel.

1. Why implement SCSI applications on Fibre Channel?

The Small Computer System Interface (SCSI)[1] is a widely accepted channel for interconnecting small computer systems and high performance peripheral devices. The Fibre Channel[2] is a newly defined channel designed to provide extremely high performance interconnections among computer systems and peripheral subsystems. Communicating across the Fibre Channel using the command set and protocol of SCSI combines the advantages of each and creates a powerful and extensible architectural model for a high performance channel. This model will be defined by the SCSI Fibre Channel Protocol (FCP), not yet published.

1.1. Properties of the Small Computer System Interface (SCSI)

The SCSI is designed to interconnect computers and intelligent peripheral devices. The peripheral devices often have high performance control processors on board that have almost as much computing power as the computer system itself. The intelligence of the peripheral devices allows the SCSI protocol to be structured as a logical communication that is common for all devices of a certain type and is independent of the manufacturer, model, and physical geometry of the device. As one example of this characteristic, a read instruction to a SCSI disk device is performed by requesting the logical block number of the first block to be read and the number of subsequent blocks to be read. No cylinder, head, or sector address is required and the mapping of the logical block number to the actual data to be read is performed by the disk device's internal processor.

The SCSI is a highly buffered interface. Peripheral devices may have several megabytes of buffering capability. While the buffering was originally designed to make the SCSI interface independent of the timing characteristics of host computers, it has had the additional effect of improving the performance of disk drives. The buffering has been used for read-ahead and write-behind data caching, for on the fly error correction, and to improve the utilization of the SCSI port of the host computer system. The peripheral devices improve the SCSI utilization by buffering up large amounts of data at the relatively low speed associated with reading from mechanical devices, then transmitting the buffered data across the interface at data rates ranging from 5 to 20 megabytes per second. Data from other devices is then interleaved on the SCSI while data transfer between the mechanical device and the buffer continues at the lower rate.

0-8194-0963-4/93/$4.00

The SCSI is a peer-to-peer interface. Any number of the devices attached to a set of SCSI cables can be host computers and any number can be peripheral devices. No switching is required for a peripheral device to be accessed by multiple hosts. The symmetry of the SCSI protocol requires a peripheral device to intelligently participate in the control of the data transfer process. The protocol even allows host computer systems to communicate with each other and peripheral devices to communicate directly with each other.

SCSI devices have been designed with very favorable cost/performance characteristics. SCSI started as a low-cost interface, but the pressures of the marketplace have raised device performance an order of magnitude and raised device capacity two orders of magnitude over the last 10 years. Each step in performance has been greeted by the demand that the cost remain unchanged, and the SCSI device designers, helped by large production volumes, have been able to meet that demand. This favorable cost/performance characteristic makes the disk drives very popular in large storage subsystems, including disk farms and Redundant Arrays of Inexpensive Disks (RAIDs).

The SCSI is a widely accepted hardware standard. Almost all desktop computers and most servers either have a SCSI interface built in to the system or have the capability of installing SCSI host adapters. The software drivers for many of those computer systems have been modularly designed to allow the easy installation of software to support new SCSI devices and to support new software functions. The modular designs typically separate the software into a host adapter driver and a target device driver. The host adapter driver manages the details of the SCSI protocol and the control of the SCSI hardware. The target device driver formulates the SCSI commands necessary to perform the functions requested by application programs. This modular structure allows the hardware implementation of the SCSI to be replaced with no changes required to the target device driver. Similarly, additional target device drivers can be added to manage other types of devices with no changes to the host adapter driver. Most companies already have a proprietary modular structure, but the proprietary structures are similar enough that a formal standard, the SCSI-2 Common Access Method[3], has been proposed to describe a standard interface between the host adapter driver and the target device driver.

The SCSI takes best advantage of its buffering, multiplexing, and command queueing capabilities when it is managed by a multi-tasking operating system. Only a multi-tasking system can activate a large number of disk devices and allow the requested operations to be multiplexed during execution and to be completed in any order. The UNIX™ operating system, commonly found on high performance desktop and server systems is an excellent example of such an operating system. UNIX™ operating systems also implement the modularity described above.

1.2. Properties of the Fibre Channel

The Fibre Channel has been defined to be a generic serial channel that provides an information delivery system between two nodes. The proposed Fibre Channel Physical and Signaling Interface standard[2] contains a description of the physical layer, including information about the connectors and driver/receiver technology. It describes the 8B10B encoding structure to be used for the transmission of bits and specifies the encoding characters that are used for structuring the data frames. It further defines a hierarchy for grouping related frames into Sequences and related Sequences into Exchanges. The basic link control functions, recovery procedures, and buffer management mechanisms are defined. The major focus of the Fibre Channel activity is to create a complete channel definition which could carry information in any required format, including SCSI.

The Fibre Channel was designed to be implemented across a wide variety of signal conductors at a wide variety of speeds, allowing the selection of the proper combination of cost and performance for any application. Data rates of 12, 25, 50, and 100 megabytes per second are defined. The signals may be carried across coaxial copper wires or multimode or single-mode optical fiber. The optical systems may use LEDs, shortwave lasers, or long-wave lasers. The early designs are modular, so that a single protocol chip can support any of several transmitter/receiver technologies. At present, Fibre Channel optical transmitter/receivers are available for long-wave single-mode operation at a 100 megabyte data rate and for shortwave multimode operation at a 25 megabyte data rate.

For architectural simplicity, the Fibre Channel is defined for symmetrical point-to-point communication between two nodes. For environments that require more than two nodes, a switching fabric is being defined that will carry the information from one node and deliver it to any of several other nodes attached to the switch. To the two nodes communicating, the port connected to the switching fabric will appear to be the port of the remote node. Some additional services are defined for support of the fabric. Three classes of service are defined between nodes. Class 1 service is defined as a dedicated connection between two nodes. Class 1 lends itself to the transfer of image data and extremely large files. There is some overhead involved in establishing the connection, but once established, the full bandwidth of the interface is available for the transfers. Class 2 service is defined as a multiplexed connectionless service with guaranteed notification of delivery between any number of nodes attached to the fabric. Class 2 service is very useful for multi-tasking operations that transfer a large number of moderate sized data files among many nodes. Such operation is typical of UNIX™ desktop and server environments. Class 3 or Datagram service is defined as a multiplexed connectionless service without guaranteed notification. Class 3 is most suitable for control and management transfers, but may also be used for the transfer of data where error recovery is managed by higher protocol layers. The Fibre Channel provides full duplex service. Data at the full bandwidth of the Fibre Channel can travel in both directions simultaneously.

1.3. Benefits of SCSI applications on Fibre Channel

The SCSI is limited to a relatively small number of devices attached over a relatively short distance. The most reasonable implementations are limited to 16 nodes with a total data rate of about 20 megabytes per second distributed along a cable 25 meters long. While this configuration is suitable for a wide range of systems, the increasing computing power of desktop and server systems makes higher channel performance a requirement. In addition, the low cost and small size of disk devices has made mass-storage subsystems a practical alternative to traditional disk configurations. The subsystems may have special capabilities, including the performance and availability improvements associated with RAID systems, automatic mirroring, and supplementary buffering. If Fibre Channel is used as the data transport channel, the data rate can be increased to 100 megabytes per second, the distances can be increased to a kilometer or more, and the number of attached subsystems or devices is limited only by the expense of the switching fabric. In addition, the small size of Fibre Channel connections and cables simplifies the attachment of the very small servers and workstations being designed today.

The Fibre Channel was defined with sufficient generality so that almost any Upper Layer Protocol (ULP) using packets or chunks of data could be transmitted across the channel. Several ULPs are presently being considered by the standardization committee, including a ULP for SCSI, the SCSI Fibre Channel Protocol (FCP). Since the SCSI command set is the most widely available mass-storage command set among high performance desktop and server systems it is likely to be a very important ULP for that environment. The layered structure of SCSI software allows the replacement of the SCSI host adapter and the host adapter driver software with a Fibre Channel host adapter and the necessary Fibre Channel host adapter driver software with very few changes to the operating system. For those systems that also require other ULPs, including the popular TCP/IP communications protocol, the Fibre Channel is designed to allow a single link to operate with multiple ULPs.

Storage subsystems may have sophisticated mappings between the high performance Fibre Channel interface and the internal disk drives. The mapping between the channel data transfer is simplest when both the channel data transfer and the disk devices use the same command set. The availability and excellent cost/performance of the SCSI disk drives makes them a logical choice for the internal disk drives and therefore makes the FCP a desirable protocol to use across the Fibre Channel.

2. How is SCSI mapped to the Fibre Channel?

The SCSI-2 standard does not make a clear distinction between the physical layer, link protocol, and command set functions. A review of the documentation indicates that the cable, connector, and signal level definitions can be separately considered as the physical layer. The link protocol includes the arbitration, selection, and reconnection phases, the requirements for sequencing the information transfer phases, and most of the messaging system. The command set functions include the division of the information transfer phases into command, data transfer, and status units as well as the actual definition of the commands. All of the physical layer and link protocol services required by the SCSI command set functions have analogs in the Fibre Channel definition. The FCP can use these Fibre Channel services to perform the required SCSI command set functions.

2.1. Description of SCSI command execution

In a SCSI system, the execution of a SCSI command requires the transfer of a command from an initiator to a target, followed by the optional unidirectional transfer of data between the initiator and the target, followed by the transfer of ending status from the target to the initiator. The complete set of information transfers associated with a single command is called an I/O Process. An I/O Process is identified in the SCSI environment by a number called a nexus. The nexus is composed of the SCSI device address of the SCSI port initiating the command, the SCSI device address of the port targeted to execute the command, and the Logical Unit Number (LUN) of the device attached to the target node. For those SCSI systems allowing the queueing of multiple commands to a single LUN, a tag is added to the nexus to differentiate the multiple I/O Processes that may be executing simultaneously between that LUN and the initiator. The nexus must be known at the physical layer and link protocol layer of the SCSI bus, but the upper layer software may use a pointer or handle in its identification of the particular I/O Process. As long as the I/O Processes can be routed between the correct devices and as long as software can uniquely identify activities associated with a particular I/O Process, the actual means of defining the nexus is not architecturally important.

The units of communication within SCSI are information transfer phases. The Command Phase transmits a packet of information from the initiator of the I/O Process to the specified logical unit. The packet contains information that defines the operation to be performed, selects the data to be used in that operation, and provides additional control parameters. After receiving the information transmitted by the Command Phase, the SCSI bus is released so that it can be used for the execution of other I/O Processes while the logical unit considers the information in the command and prepares itself to perform the required logical operation. When the logical unit is ready to perform any required data transfers, it requests reconnection to the SCSI bus and begins the transfer. Read operations move data from the logical unit to a buffer in the target, then, when the reconnection is successful, transfer the data from the target to the initiator using a Data In Phase. Write operations prepare a buffer in the target, reconnect and transfer data from the initiator to the target buffer using a Data Out Phase, then move the data from the target's buffer to the logical unit. When all data movement is done and data transfer to or from the logical unit has been verified, a packet of information describing the normal or abnormal completion of the I/O Process is transferred from the logical unit to the initiator using the Status Phase.

2.2. Description of Fibre Channel functions available to an Upper Layer Protocol (ULP)

While the Fibre Channel standard defines a physical interface and an encoded-word interface, the interface that is interesting for the FCP is the FC-PH data service interface. The interface describes a hierarchy of data units. The largest data unit is the Exchange, composed of a group of one or more Sequences. Each Sequence is composed of one or more frames. The frame is defined as the minimum unit of information transfer which can be directed from one node to another. The frame contains a header that completely describes the source of the frame, the destination of the frame, the ULP of the frame, and the type of information contained within the frame. The frame's header also describes the frame's membership in an Exchange and Sequence and describes the position of the frame within a Sequence. Each frame transmission is verified by an acknowledgment returned to the device transmitting the frame.

A Sequence is a unidirectional set of data composed of one or more frames. Only one Sequence is allowed at a time for each Exchange. The ordering and size of the frames in the Sequence is not important, since the smallest unit of data that can be manipulated at the service interface is the complete Sequence. ULPs perform data recovery at the Sequence level.

The Exchange is a set of Sequences with a relationship defined by the ULP. Sequence numbering is established by the Fibre Channel within an Exchange. There may be an arbitrary number of Exchanges transferring information in both directions between two nodes, depending on the class of service. Each Exchange is uniquely identified by an Originator Exchange ID, offered by the node that starts the Exchange, and a Responder Exchange ID, provided by the node that services the Exchange. The Exchange ID's are unique between a pair of nodes. The combination of Source Identifier, Destination Identifier, Originator Exchange ID and Responder Exchange ID defines an Exchange uniquely. Additional ULP addressing may be required to identify the logical unit associated with that Exchange.

An architectural concept called the Information Unit is presently being discussed by the standardization committee as a method of describing groupings of Sequences within an Exchange. This data unit is not a necessary concept for the SCSI mapping and is still being strongly discussed by the X3T9.2 and X3T9.3 standard committees.

Link management, including Fabric Login, Node Login, and the Exchange of link management parameters, is performed using Link Control frames. Information in the header differentiates Link Control frames from Link Data frames.

2.3. Mapping of SCSI functions to Fibre Channel

The mapping of SCSI information to the Fibre Channel is very simple. The node addresses and Exchange IDs uniquely define an Exchange in the same manner that the nexus uniquely defines a SCSI I/O Process.

A Sequence within a Fibre Channel Exchange has the same characteristics as a SCSI Phase within an I/O Process. The Sequence is unidirectional. There must be only one Sequence being transmitted at a time within an Exchange, but multiplexed transmission of the frames of Sequences for different Exchanges is allowed. The contents of a Sequence are uniquely defined by fields within the headers of frames in the Sequence.

A Fibre Channel frame is an architectural convenience. The data being carried in a Sequence is divided into frames to allow conveniently small groups of data to be transmitted by the hardware. The header of the each frame provides an offset value and a frame count so that the frame can be located correctly in the stream of data being transmitted within the Sequence. The frame size, ordering requirements, and other transmission characteristics are agreed upon by the communicating nodes during the Fibre Channel login operations.

Table 1 shows the mapping between the Fibre Channel architectural constructs and the SCSI architectural constructs.

SCSI construct	FC construct
Architectural Concepts:	
IO Process	Exchange
Phase	Sequence
Phases:	
Command Transfer	Command Sequence
Reconnection Pointers	Transfer Ready Sequence
Data Transfer	Data Transfer Sequence
Status Transfer	Response Sequence

Table 1: Relationship between SCSI and Fibre Channel

2.4. Typical sequencing of a SCSI I/O Process on Fibre Channel

Sequences of a Fibre Channel are defined according to a convention that describes the properties of those Sequences. Table 2 and 3 describe the Sequences used by the FCP. Each Sequence is given a sequence name as a reference to its properties. Those beginning with an I are transmitted from the initiator (the Fibre Channel originator), while those beginning with a T are transmitted from the target (the Fibre Channel responder).

Table 2: SCSI SEQUENCES SENT TO TARGETS

#	PHASE	DATA BLOCK		F/M/L	SI	SC	M/O	COMMENTS
		CAT.	CONTENT					
I1	CMD	2	COMMAND	F	T	0	M	SCSI CMD
I2	DATA	1	DATA	M	T	0	M	ONLY DATA

Table 3: SCSI SEQUENCES SENT TO INITIATORS

#	PHASE	DATA BLOCK		F/M/L	SI	SC	M/O	COMMENTS
		CAT.	CONTENT					
T1	DATA	5	XFR RDY	M	T	0	M	WRITE XFR RDY
T2	DATA	5	XFR RDY	M	H	0	M	READ XFR RDY
T3	DATA	1	DATA	M	H	0	M	ONLY DATA
T4	STATUS	3	STATUS	L	T	0	M	STATUS BLOCK

Key:

#	Sequence Name
CAT.	Information Category of data block
CONTENT	Contents (Payload) of data block
F/M/L	First/Middle/Last sequence of exchange
SI	Sequence Initiative--Held or Transferred
SC	Sequence Count (SEQ_CNT)--Start from 0 or Continued
M/O	Mandatory/Optional Sequence
XFR RDY	Transfer Ready

The I1 Sequence contains a packet of unsolicited control information describing the SCSI command. The packet is also the first Sequence of the Exchange and defines the beginning of the SCSI I/O Process. At the end of the Sequence, initiative is transferred to the target, requiring the next Sequence of the Exchange to be provided by the target.

The T1 and T2 Sequences contain a special packet that defines the length of the data transfer to be made and indicates that the target is prepared for the transfer. The Sequence either retains or transfers initiative depending on whether the operation is a read from the target or a write to the target.

The I2 and T3 Sequences contain the solicited data to be transferred to or from the target. The Data Sequence can be arbitrarily long.

The T4 Sequence contains the solicited control information indicating the successful or unsuccessful execution of the requested command. The status block contains any SCSI Request Sense information that results from the unsuccessful execution of the requested command. The T4 Sequence also is the final Sequence of the Fibre Channel Exchange and marks the completion of the I/O Process.

Typical read and write sequences are shown in Figures 1 and 2.

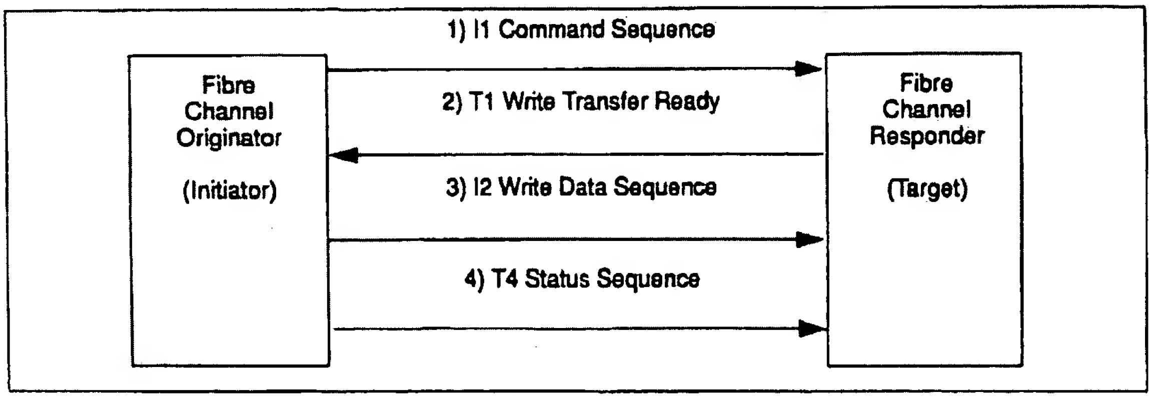


Figure 1: Example of typical write operation using FCP

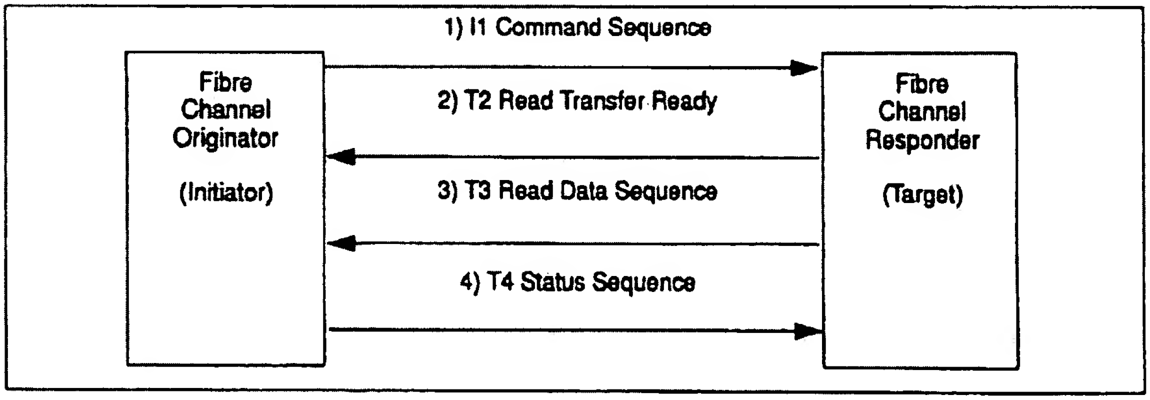


Figure 2: Example of typical read operation using FCP

2.5. Content of information packets

2.5.1 Command Sequence

The I1 Command Sequence carries all the command information necessary to execute a SCSI I/O Process. In the SCSI protocol, most of that information is carried by the Command Descriptor Block (CDB) which is transmitted during the Command Phase. Some additional information is carried implicitly or explicitly in the link protocol. The FCP Command Sequence must carry both the CDB and the implicit information. Table 4 describes the information carried by the FCP Command Sequence.

FCP Command Field	Length (Bytes)	Description
SCSI_ENT_ADDR	8	Additional addressing information to define real or virtual logical unit
SCSI_CNTL	4	Control Flags
SCSI_DL	4	Length of Data Transfer to be performed
SCSI_CDB	16	Command Descriptor Block defined by SCSI

Table 4: Contents of FCP Command Sequence

The SCSI Entity Address is a supplementary address structure that allows the identification of any device within an attached subsystem. The structure allows the grouping of one or more physical devices into various virtual logical units. The Entity Address allows the groupings to be organized in a hierarchical manner, allowing both the virtual logical units and the managing control units to be directly addressed.

The SCSI Control field contains control flags to perform Command Phase functions implemented by the SCSI link protocol. These control flags are outlined in table 5.

Flag	Description
Abort	Requests termination of all I/O Processes for addressed entity and below. Response returned for each terminated process.
Abort Tag	Requests termination of selected I/O Process. Response returned.
Reset	Resets addressed entity and below. No response for terminated I/O Processes.
Read Data	Read data transfer expected.
Write Data	Write data transfer expected
Head_of_Q	I/O Process should be executed next.
Ordered_Q	I/O Process should be executed after all previous and before all subsequent I/O Processes
Simple_Q	I/O Process executed at any time.

Table 5: Contents of FCP Command Sequence Control Flags

The SCSI Data Length specifies the length of data transfer to be performed. This information is implicit in the SCSI link protocol, but must be carried explicitly in the FCP.

The SCSI Command Descriptor Block defines the actual SCSI I/O Process to be executed.

2.5.2 Transfer Ready Sequence

The information carried by the T1 and T2 Transfer Ready sequence consists of a four byte length field defining the amount of data that will be transferred in the following Data Sequence. The T1 Write Transfer Ready indicates that buffering has been dynamically allocated for the specified amount of data to be transferred from the initiator. The T2 Read Transfer Ready warns the initiator to expect to receive the specified amount of data. Most FCP data transfers will be completed in a single Data Transfer Sequence. Extremely large transfers, transfers of unknown maximum length, transfers using limited buffering, or transfers with long latencies may be accommodated by using more than one set of paired Transfer Ready and Data Transfer Sequences.

2.5.3 Data Transfer Sequence

The data transfer sequences, I2 and T3, contain data to be transferred to and from the destination logical unit. For some SCSI CDB's, the data actually contains control parameters or state information associated with the destination logical unit.

2.5.4 Response Sequence

The Response Sequence, T4, contains the SCSI Status byte. Any error information related to the particular I/O Process is also returned as part of the Response Sequence. The inclusion of the error information as part of the I/O Process avoids the difficulties of associating separated error reports with the failing process. The contents of the Response Sequence are shown in table 6.

Field	Size (Bytes)
SCSI_STATUS	4
SCSI_RESID	4
SCSI_SNS_LEN	4
SCSI_RSP_LEN	4
SCSI_SNS_INFO	n
SCSI_RSP_INFO	m

Table 6: Response Sequence Contents

The SCSI Residual Length field is the difference between the number of bytes that the were expected to be transferred and the number of bytes actually transferred in the Data Transfer Sequences. The number of bytes expected to be transferred is obtained from the Data Length field in the Command Sequence.

The information normally provided by a SCSI Request Sense command is contained in the SCSI Sense Information field. Since this field is not of a fixed size, a descriptive length field is also provided.

Failures associated with the Fibre Channel link and the supporting responder hardware that are not reported through the normal Fibre Channel error presentation mechanisms are reported in the SCSI Response Information field. This may include FCP protocol violations as well as hardware errors. There is not yet a standard definition of the SCSI Response Information.

3. Conclusions

The SCSI Fibre Channel Protocol allows each of the standard architectures, SCSI and Fibre Channel, to use the best attributes of the other. The Fibre Channel standard provides a very high bandwidth SCSI connection to a very large number of nodes over relatively great distances. The Fibre Channel provides a variety of cost and performance options for the execution of SCSI I/O Processes. The SCSI provides a standard and popular software interface as a Fibre Channel communication model between computer systems and a wide variety of storage subsystems and peripheral devices. The marketplace appears to be quickly taking advantage of this synergy.

4. Acknowledgments

The architectural work that forms the basis of this paper is a contribution to the X3T9.2 Task Group of the Computer and Business Equipment Manufacturers Association. The author is supported in this activity by Sun Microsystems Computer Corporation.

5. References

1. Secretariat, Computer and Business Equipment Manufacturers Association, *Small Computer System Interface - 2 (SCSI-2), X3T9/89 042, revision 10h*, October, 1991.

2. Secretariat, Computer and Business Equipment Manufacturers Association, *Fibre Channel Physical and Signaling Interface (FC-PH), Revision 3.0, X3T9/91-062*, June, 1992.

3. Secretariat, Computer and Business Equipment Manufacturers Association, *SCSI-2 Common Access Method, Transport and SCSI Interface Module, Revision 3.0*, April, 1992

Since the above documents are still in the ANSI standardization process, they are not yet available from the American National Standards Institute. They are available from Global Engineering, 2805 McGaw St. Irvine, CA 92714, (800) 854-7179.

INTERNATIONAL SEARCH REPORT

ational Application No
PCT/US 00/42337

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L12/24

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L H04Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 767 427 A (DIGITAL EQUIPMENT CORP) 9 April 1997 (1997-04-09)	1,8,13
Y	page 6, line 27 -page 8, line 15 page 28, line 15 - line 57 ---	2-4,6, 9-11, 14-16

-/--

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search 25 September 2001	Date of mailing of the international search report 09/10/2001
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Peeters, D

INTERNATIONAL SEARCH REPORT

ational Application No
PCT/US 00/42337

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L12/24

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L H04Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 767 427 A (DIGITAL EQUIPMENT CORP) 9 April 1997 (1997-04-09)	1,8,13
Y	page 6, line 27 -page 8, line 15 page 28, line 15 - line 57 ---	2-4,6, 9-11, 14-16

-/--

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search 25 September 2001	Date of mailing of the international search report 09/10/2001
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Peeters, D

INTERNATIONAL SEARCH REPORT

ational Application No
PCT/US 00/42337

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5 559 958 A (DIDNER JONATHAN R ET AL) 24 September 1996 (1996-09-24) column 1, line 49 -column 3, line 33; figure 1 column 5, line 4 - line 44 column 8, line 14 - line 50; figure 2 column 9, line 50 - line 67; figure 2 column 33, line 34 -column 34, line 13; figure 6A column 200, line 1 - line 11; figures 9C,9D,13 column 202, line 5 - line 28; figures 13,14 column 212, line 29 - line 54; figures 9A,10-15 ---	2-4,6, 9-11, 14-16
A	DICK BANNISTER: "Compaq Storage Resource Manager" STORAGE NEWS - EVALUATOR GROUP, 'Online! vol. 2, no. 4, April 2000 (2000-04), XP002178293 Retrieved from the Internet: <URL:http://www.evaluatorgroup.com/English/Collaterals/Newsletter/200004_Newsletter.pdf> 'retrieved on 2001-09-24! page 5, right-hand column page 6, left-hand column -right-hand column -----	1-17

INTERNATIONAL SEARCH REPORT

Information on patent family members

ational Application No
PCT/US 00/42337

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0767427 A	09-04-1997	US 5345587 A	06-09-1994
		EP 0767427 A2	09-04-1997
		AT 160034 T	15-11-1997
		AU 3980093 A	19-08-1993
		AU 3980193 A	19-08-1993
		AU 3980293 A	05-08-1993
		AU 3980393 A	19-08-1993
		AU 3980493 A	05-08-1993
		AU 639416 B2	29-07-1993
		AU 4305289 A	02-04-1990
		DE 68928433 D1	11-12-1997
		DE 68928433 T2	16-04-1998
		EP 0441798 A1	22-03-1990
		JP 6502505 T	17-03-1994
		WO 9003005 A1	22-03-1990
		US 5475838 A	12-12-1995
		US 5557796 A	17-09-1996
		US 5608907 A	04-03-1997
		US 5832224 A	03-11-1998
		CN 1043810 A	11-07-1990
		JP 2230847 A	13-09-1990
		CN 1044176 A	25-07-1990
		CN 1044175 A	25-07-1990
		JP 2277153 A	13-11-1990
		CN 1045656 A	26-09-1990
		CN 1044174 A	25-07-1990
		JP 2236767 A	19-09-1990
		CN 1044719 A	15-08-1990
		JP 3062155 A	18-03-1991
US 5559958 A	24-09-1996	US 5471617 A	28-11-1995
		US 5828583 A	27-10-1998
		AT 198115 T	15-12-2000
		CA 2104421 A1	22-02-1994
		DE 69329743 D1	18-01-2001
		DE 69329743 T2	12-04-2001
		EP 0585082 A2	02-03-1994
		JP 2533066 B2	11-09-1996
		JP 6175955 A	24-06-1994
		AT 164242 T	15-04-1998
		CA 2071804 A1	25-12-1992
		DE 69224775 D1	23-04-1998
		DE 69224775 T2	06-08-1998
		EP 0520769 A2	30-12-1992
		JP 5257914 A	08-10-1993
		US 5367670 A	22-11-1994

AS8

Evaluator Group

STORAGE NEWS

Volume 2, Issue 4

April 2000

INSIDE THIS ISSUE:

New Publication

We have added a new publication to our existing portfolio - the StorageTek 9500. This is the latest product from the Iceberg stable.

Our publications provide in-depth product information and our commentary. At just $95 a copy these are a bargain.

Publications can be ordered by credit card from our web site.

Dick Bannister

IBM'S SAN INITATIVE

Stating the industry need to harness the explosive growth in e-commerce, IBM announced a $400 million SAN initiative. With this SAN initiative, IBM is leveraging its position as the only company that has the products, open systems capability, services and systems management experience to provide heterogeneous SAN solutions.

IBM announced new products, services, global and regional testing facilities, plus a significant expansion of its sales force to provide SAN solutions based on industry standards. "IBM's SAN initiative is about leveraging every division of the company to deliver on the promise of SANs for our customers; interoperability between vendor systems and true data sharing," said Linda Sanford, General Manager, IBM Storage Subsystems Division. "Any industry vendor not committed to open standards for SANs is driving their customers into a proprietary dead-end."

Highlights

The following are the highlights of the March 28, 2000 announcement:

- IBM is establishing more than 50 SAN Solution Centers with IBM Business Partners worldwide to develop SAN solutions and demonstrate them to customers.
- New SAN testing facilities in Montpellier, France and Makuhari, Japan.
- Expansion of IBM's storage solutions sales force with more than 1,000 additional sales specialists.
- Establishment of SAN and Storage Services consulting practice by IBM Global Services.
- New models of the 2105 (Shark) Enterprise Storage Server, offering up to a 100 percent improvement in performance.
- New Fibre Channel SAN solutions, including fabric support for Netfinity servers.

IBM Global Services is offering a portfolio of new services to aid customers implement SANs and other storage solutions. They include the following:

- **Enterprise Storage Assessment** - review customers existing storage strategy and identify storage business needs and requirements.
- **Planning and Design Services** - build a complete storage and SAN solution using a variety of industry standard technologies.
- **Implementation Services** - conduct proof-of-concept storage and SAN tests, evaluate SAN solutions and assist with the SAN implementation.

(continued on page 6)

Copyright 2000 Evaluator Group Inc. All rights reserved

ES/OL DATABASE

Buying storage has become more complicated and more time consuming due to:

- Consolidating storage from multiple servers
- Planning or implementing Storage Area Networks
- Multi-platform storage products
- A broader range of products

In the ideal world, to ensure due diligence, when buying storage you need to:

- Get briefed by all the storage and switch vendors
- Peruse their web sites
- Eliminate the marketing hype
- Evaluate each offering for strengths and weaknesses
- Compare SAN strategies
- Put all this information into a common format
- Produce matrix comparison charts
- Discuss options with industry experts

And when you have completed those tasks - so much time has passed that your data needs updating so you need to start again! Or you can save time, do a more effective and efficient job by subscribing to our *ES/OL* database. The annual subscription price for non-vendors is considerably less than for vendors, and includes unlimited telephone consultation, corporate wide access and a free seat at a seminar.

Briefly what our ES/OL database provides is:

- A web accessible database of current information on open and mainframe disk subsystems, related software and SAN switch information
- A directory of the last 30 days of announcements with a hot link to more detailed information
- A matrix charts comparing functions by vendor in various categories
- Company overview, financial, marketing and SAN strategy and three months of announcement overviews
- Product history, product overview, tech specs, product software, list prices
- Product overviews that include our perception of strengths and weaknesses and "gotcha's"
- Vendors covered: Amdahl, Compaq, Dell EMC (CLARiiON), HDS, HP, IBM, LSI, Sun, StorageTek and eleven SAN switch vendors
- Detailed Announcement Analysis of major product announcements
- And we update all the information daily

Our clients for the database include most of the vendors and a wide range of end users from Ameritech to Zurich Insurance.

We provide a level of detail that the other analysts do not. We provide information that helps the "implementors" rather than the strategists. Please compare our content and our price with the other analysts.

If you would like a guided tour of our ES/OL database - please let us know.

ES/OL Database— Information at your fingertips

Copyright 2000 Evaluator Group Inc. All rights reserved

SAN GLOSSARY—CONTINUED

Striping A method for achieving higher bandwidth using multiple N_ports in parallel to transmit a single information unit across multiple levels.

Switch A device that provides a crossbar connection such that any one port may be connected to any other port at any one instant such that multiple connections are occurring.

Synchronous Optical Network (SONET) A standard for optical network elements providing modular building blocks, fixed overheads and integrated operations channels, and flexible payload mappings. Basic level is 51.840 megabit/sec (OC-1); higher levels are n times the basic rate (OC-n).

A technical committee of the National Committee for Information Technology Standards, titled T11 I/O Interfaces. It is tasked with developing standards for moving data in and out of central computers. (also known as X3T11)

Time Server A Fibre Channel-defined service function that allows for the management of all timers used within a Fibre Channel system.

Topology The logical and/or physical arrangement of stations on a network.

Transmitter A device that includes a source of driving elements.

TCP/IP Transmission Control Protocol over Internet Protocol.

ULP Upper Layer Protocol.

UNIX Generic term for the various implementations of the UNIX operating system.

Virtual circuit A unidirectional path between two communicating N_Ports that permits fractional bandwidth.

WWN (World Wide Name) A unique name for a fibre channel port that is assigned to a manufacturer by the administering body, the IEEE.

Zone A logical (or physical blockage creating a grouping of ports in a switch that is allowed to communicate with each other. Other elements not in the zone cannot communicate.

IBM and McData Reseller Agreement

IBM and McDATA Corporation announced a worldwide reseller agreement under which IBM will resell the McDATA ED-5000 fiber channel switch and management software (under the McDATA logo). The ED-5000 is an enterprise fiber channel director (fabric switch) and is patterned after the currently available ESCON Directors provided by IBM.

The McDATA ED-5000 Fibre Channel Director provides enterprise-class availability, connectivity, and performance features. These features include the following:

- Up to 32 ports with any-to-any connectivity
- A total aggregate bandwidth of 3,200 MB/second
- Centralized management
- Complete redundancy of all active support elements with automatic failover
- Hot-swap capability for all major field-replaceable components

(continued on page 4)

Copyright 2000 Evaluator Group Inc. All rights reserved

(continued from page 3)

- Non-disruptive service, upgrades, and code changes
- End-to-end data integrity checking with intelligent fault isolation

Note: For our ES/OL clients please reference the *ES/OL* SAN Components for more information.

Evaluator Group Comments:

The addition of an enterprise-class director was previewed by IBM in its SAN roadmap last year. IBM has had a long relationship with McDATA with the ESCON Director making the ED-5000 the logical choice. We believe this is a win-win relationship for both companies. IBM adds an enterprise fibre channel switch to its SAN components and McDATA benefits from partnering with the world's largest computer manufacturer. This relationship will also benefit the industry as interoperability between Brocade and McDATA can be guaranteed.

Enhancements to Compaq Data Replication Manager

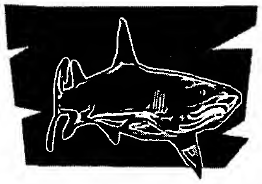
The Compaq SANworks Data Replication Manager now supports:

- ATM over a single T1/E1 wide area network
- ATM over multiple T1/E1 wide area network (implementing Inverse Multiplexing)
- ATM over T3/E3 wide area network
- ATM over Fractional and/or shared T3/E3 and OC3 wide area network
- Fibre Channel connection via Wave Division Multiplexing (WDM) and Dense Wave Division Multiplexing (DWDM)

The enhanced connectivity allows for greater flexibility in implementing remote copy solutions. Wide Area Network (WAN) connections over varying distances at maximum telephony speeds gives IT operations choices based on business needs and costs. The addition of fractional bandwidth capability with shared ATM or WDM (and DWDM) are again more choices for cost and performance tradeoffs.

Evaluator Group Comments

The Data Replication Manager is an excellent peer-to-peer remote copy solution that has now been greatly enhanced with more connectivity options



The Good, The Bad and The Indifferent

In March, IBM announced new models of the 2105 Enterprise Storage Server (ESS), Enhancements, (Good), Slippages (Bad) but these slippages may be indifferent to some users.

The new 2105-F10 and –F20 models have additional busses, faster RISC processors and up to 16GB of cache. The size of the non-volatile storage (NVS) remains at 384MB. ***The increased cache size should enable customers to take greater advantage of the scalability of the ESS. Even though IBM has stated that the limitation of 384MB of NVS has not been a problem due to the speed of the SSA backend architecture, we still have some concerns about this area for environments with high random write activity.***

A new LIC, which IBM refers to internally as 1.1, is planned for general availability in April. It is considered as the "base" LIC and is required for support of advanced functions, including XRC, PPRC, FlashCopy and native Fibre Channel. Existing Model E10 and E20 customers with LIC 1.0 will have to install this new microcode to support these advanced functions, **and the installation is disruptive.**

The IBM Subsystem Device Driver is a new, no charge addition to the ESS. It provides path failover and load balancing for either Fibre Channel or SCSI connected Windows NT, HP-UX, Sun Solaris and AIX platforms. IBM states that it is mandatory for these platforms.

Copyright 2000 Evaluator Group Inc. All rights reserved

SHARK (continued from page 4)

The availability of XRC, originally scheduled for the first quarter of 2000, has also been delayed.
XRC with planned outage support will be available as originally scheduled on March 31st. However, support for XRC with unplanned outage support (Suspend/Resume) has been delayed until the summer of 2000.

The availability of PPRC for all platforms, originally scheduled for the first quarter of 2000, has been delayed until the fall of 2000.

The availability of FlashCopy, originally scheduled for the first quarter of 2000, has been delayed until the summer of 2000 for OS/390 and the fall of 2000 for all other platforms.

Native Fibre Channel support, originally scheduled for first quarter, has also been delayed. Depending upon the host platform, the planned availability dates are now the summer or fall of 2000. Until Native support is shipped, IBM announced interim Fibre Channel Host Adapters.

Summary

IBM's success with the ESS since it became generally available in September 1999 had been outstanding, with approximately 1,500 installed subsystems. The product has demonstrated excellent performance characteristics, especially in OS/390 environments due to the unique Parallel Access Volume (PAV) and Multiple Allegiance (MA) features.

However, when the ESS was first introduced, we stated that the long-term success of this product in competing with the EMC Symmetrix, Hitachi Data Systems 7700E, and HP XP256, would depend upon IBM's ability to deliver advanced features and functions, such as PPRC, FlashCopy and native Fibre Channel, in a timely manner. This announcement of delays in the availability of these critical features is a major blow to this product. It will not be a viable alternative for customers with an immediate requirement for these advanced features, especially PPRC and FlashCopy. They will be forced to wait or choose one of the other alternatives and implement the associated remote copy and/or point-in-time copy solutions. Once these unique solutions are implemented and become imbedded in operational procedures, the customer will probably become locked into these competitive products for the next two to three years.

Compaq Storage Resource Manager

In March, Compaq announced their SANworks Storage Resource Manager. This product is the result of Compaq's strategic partnership with HighGround Systems that was announced on December 6, 1999. The Storage Resource Manager is a tool to monitor and report on storage connected to servers via a SAN. The monitoring software collects data and stores it in a database on the server running the SRM. The information is used for detecting trends, providing web-based event management and alerts and providing reports. With this announcement, Compaq has delivered one of its key components of the Enterprise Storage Network Architecture (ENSA): storage resource management.

Overview

The Storage Resource Manager is a tool with a server component and agents that run on other servers that have ownership of storage. The server where Storage Resource Manager executes must be running Windows NT on an Intel compatible processor with the following software:

- Microsoft Windows NT Server 4.0 SP3 or later
- Microsoft Internet Explorer 4.01 or later
- Microsoft Internet Information Server 4.0
- Microsoft SQL Server 6.5 SP4 or SP5 or Server 7.0

Airline Humor

actual quotes

After landing : "Thank you for flying Delta Business Express. We hope you enjoyed giving us the business as much as we enjoyed taking you for a ride."

- BM also announced enhanced fiber channel

Copyright 2000 Evaluator Group Inc. All rights reserved

IBM's SAN Initiative (continued from page 1)

- **Support Services**—help customers operate their storage and SAN configurations, including troubleshooting services and problem resolution, as well as upgrades and solution management.

- **SAN Testing Facilities** - IBM will open new SAN testing facilities in Montpellier, France and Makuhari, Japan that allow customers to evaluate and prototype SAN solutions for reliability, performance and security. This facilities are additions to the test facilities located in Gaithersburg, Maryland and Mainz, Germany.

IBM also introduced a SAN solution using a cluster of two Netfinity servers with fiber channel connections and Legato mirroring extension software. The cluster can be separated up to ten kilometers.

Evaluator Group Comments:

IBM is indeed flexing its size and depth in the SAN marketplace. The interoperability testing facilities and professional services indicate IBM's commitment to the SAN market.

Key for IBM will be the roll-out of the Tivoli SAN Storage Manager. The introduction of the SAN Storage Manager, based on the Mercury SANergy, has the potential to accelerate SAN solutions and place IBM as the leader in SAN implementations.

Storage Resource Manager for Exchange

This product is another result of Compaq's strategic partnership with HighGround Systems that was announced on December 6, 1999 and is related to its SANworks Storage Resource Manager (see page 5). This product addresses a specific Storage Resource Management application that cannot be handled in the general SRM case. This is a tool to monitor and report on Microsoft Exchange information stores. The monitoring software collects data and takes policy driven action on the data on the server running the SRM. The information is used by an administrator to prevent interruption of service with Exchange due to out-of-space conditions and for management of storage within Exchange.

Overview

The Storage Resource Manager for Exchange has a server component and agents that run on other servers that are running Microsoft Exchange. The server where Storage Resource Manager for Exchange executes must be running Windows NT on an Intel compatible processor with the following software:

- Microsoft Windows NT Server 4.0 SP3 or later
- Microsoft Internet Explorer 4.01 or later
- Microsoft Internet Information Server 4.0
- Microsoft SQL Server 6.5 SP4 or SP5 or Server 7.0

The SRM server will monitor Exchange storage that is owned by other servers if they are running agent software. Agents are available for the following operating systems:

- Windows NT Server / Workstation 4.0 with SP3 or later
- Windows 2000 Server / Advanced Server

By interacting with the servers running the agents over TCP/IP, the SRM Exchange server collects information, reports on conditions with alerts, and invokes actions based on policies. Access to the information on the server is via the web through LAN, WAN, or dial-up connection. The alerts that are generated based on threshold settings can be issued via browser, e-mail, and SNMP consoles. Reports may be generated from the collected data.

Evaluator Group Comments

This announcement provides a point product for resource management that specifically addresses the NT/2000 marketplace. This will be a useful tool for administrators wanting to consolidate resource management for Exchange.

Copyright 2000 Evaluator Group Inc. All rights reserved

Evaluator Group

7720 East Belleview Ave
Suite 210
Englewood, CO 80111
USA

Phone: 303-221 7867
Fax: 303-221-1615
Email: info@evaluatorgroup.com

We're on the web!
www.evaluatorgroup.com

Evaluator Group, Inc.
Comprehensive Information Storage Evaluation

Go to www.cjta.net/special/index.htm. If you cannot figure it out—send me an email for the answer.

Want to find out about Network attached Storage and how it compares to SAN? We have a free article on our web site that helps to explain the differences and where each is appropriate.

Dick Bannister

New Seminar Dates

A number of our international participants have a need to come to both our SAN Seminar and the Multi Platform Seminar. But because these seminars are separated by at least a month, this requires two trips to the United States. This is invariably too much for their budgets. So, to accommodate our foreign clients, we are changing our next seminar dates to make them contiguous.

The Next Multi Platform Seminar will be June 11– 12 immediately followed by the SAN Seminar on June 13-14. They will be held in Denver and registration and logistical information is on our web site.

The Multi Platform seminar is for those users who plan to consolidate both open mainframe storage on the same subsystem. Currently, Amdahl, EMC, Hitachi Data Systems, HP, IBM and StorageTek all have offerings. We explain their capabilities and point out their strengths and weaknesses.

"This was brilliant! It is very difficult to obtain information presented in an objective manner. The fact that you pointed out both the pros and the cons of these systems will allow for much better decision making."

Our SAN seminar is for accounts planning to, or considering a SAN implementation. There are many claims from the various vendors. It is difficult to determine what is marketing hype and how do you prevent yourself being locked into a single vendor? What are the various vendor SAN strategies? Whose ahead in SAN plans and SAN implementation? These and many other questions are answered in this seminar. You leave with copies of over 400 foils used in the seminar. We restrict the class size to encourage audience participation.

"An amazingly comprehensive collection of key industry information synthesized from many different sources...all of which I could do myself, if I had a hope in h... of finding enough time to do the research!! The value equation returns high marks for these presentations."

ES/OL DATABASE GROWS

- We have now added: Atto, Chaparral, Crossroads and Pathlight, bringing to eleven, the Fibre Channel Components vendors we monitor and provide information on.

- We have added an archival section for announcement information.

- We will be adding SAN management software

- We will be adding Network Attached Storage products

Copyright 2000 Evaluator Group Inc. All rights reserved

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

□ **BLACK BORDERS**

□ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

□ **FADED TEXT OR DRAWING**

□ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

□ **SKEWED/SLANTED IMAGES**

□ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

□ **GRAY SCALE DOCUMENTS**

□ **LINES OR MARKS ON ORIGINAL DOCUMENT**

□ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

□ **OTHER:** ______________________________

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.